Joint Live Virtual and Constructive (JLVC) Federation Integration Guide

United States Joint Forces Command (USJFCOM)

Technical Development and Innovation Branch (TDIB)

Version 3.1

13 January 2010

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
13 JAN 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Joint Live Virtual and Constructive (JLVC) Federation Integration Guide	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **United States Joint Forces Command (USJFCOM),1562 Mitscher Ave. Suite 200,Norfolk,VA,23551**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**166**	

Record of Revisions and/or Changes		
Document Revision No.	**Revision Date**	**Description of Change**
1.0	24 January 2009	JLVC Federation Integration Guide Version 1.0
3.1	18 December 2009	Page i, cover sheet, Renumbered revision to correspond to JLVC version number Page 2-5, Section 2.4, Addition of Figure 1, JLVC Conceptual Model Page 4-2, Section 4.2.1, Addition of introductory paragraph Page 4-9, Section 4.3, Re-worded first paragraph Page 4-11, Section 4.4.1, Removed link to OBS Schema, added reference to, and included OBS Schema in, Appendix D Page 5-2, Section 5.1.1, Removed link to FOM, added contact information for obtaining FOM. Pages 5-3 Section 5.1.2.2, Corrected numbering in sub-sections Pages 5-3 and 5-4, Section 5.1.2.3, Numbered sub-sections Page 5-5, Section 5.1.2.4, Added “[Delete Object Instance] should be used for Munition objects. All other object type instances should exist for the duration of federation execution.” Pages 5-7 and 5-8, Section 5.1.2.6.1, Numbered sub-sections Page 5-25, Section 5.1.7, Added MTWS sim name and site ID Page 6-1, Section 6.1, Revised wording Pages C-1 and C-2, Appendix C, Updated revision information Pages D-1 thru D-16, Appendix D, Added OBS Schema and moved Software – Event Questionnaire to Appendix H Pages E-1 thru E-65, Appendix E, Added Object Class Publication and Subscription sets Pages F-1 thru F-2, Appendix F, Added Interaction Class Initiate and Receive sets Page G-1, Appendix G, References a separately published Federation Object Model Pages H-1 thru H-2, Appendix H, Moved Software – Event Questionnaire from Appendix D

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

APPENDICES

1.0 Introduction

1.1 Purpose

The Joint Live, Virtual and Constructive (JLVC) Federation Integration Guide describes the JLVC federation from the perspective of federate developers and federation integrators. The Integration Guide introduces the reader to key JLVC architectures and interfaces and directs the reader to authoritative technical documents for implementation details. This guide is not intended to be the sole reference source in the development of JLVC-interoperable federates, but rather, to augment technical interchanges in a collaborative engineering integration environment between JLVC modeling and simulation (M&S) Engineering Team and federate developers in the following tasks:

- Estimate the effort required to integrate a federate with a JLVC Federation.
- Develop federates that integrate correctly with the JLVC Federation.
- Introduce new JLVC developers to the JLVC architecture, key interfaces, and standard implementation practices.

1.2 Document Changes/Updates

This document contains approved guidelines for the use of the JLVC for joint training and other uses as approved on a case-by-case basis by the JLVC Technical Director.

Any changes to this document will be recorded in a change record. When a new version of the document is issued, as indicated by the number following the “V” in the document number, the previous version is automatically superseded.

1.3 Document Organization

This document is organized into nine sections. Sections 1 through 4 provide a background and overview on JLVC Federation development and use. Section 5 includes the technical specifications needed to operate in the Federation. Section 6 is the governance portion, Section 7 addresses configuration management, and the Appendices lay out the operational specifications by functional area. Here is a brief description of all the sections:

- Section 1 describes the purpose of the document and its organization
- Section 2 defines the terms live, virtual and constructive and describes the JLVC purpose and objectives. It also presents suggested uses of the JLVC components to support Commanders’ training strategies to achieve and sustain command, unit and staff proficiency on selected Mission Essential Task List (METL) tasks and supporting unit and staff battle tasks. Section 2 also provides typical use cases, and identifies the benefits and drawbacks of using the JLVC

environment to support exercises. This section follows the guidance contained in Federation Development and Execution Process (FEDEP) documents by discussing the conceptual model and high-level requirements.

- Section 3 continues to follow the FEDEP by describing functional allocation among federates.
- Section 4 describes the JLVC architectures to include the Command, Control, Communications, Computers, and Intelligence (C4I) and simulation architectures. It also discusses the Joint Training Data Services (JTDS).
- Section 5 begins the more technical portion of the document. This section discusses the federation agreements by describing the JLVC interface standards to include the Federation Object Model (FOM), Run-time Infrastructure (RTI), and model characteristics dealing with attrition, consumption, resurrection, and resigning.
- Section 6 contains organizational information regarding the JLVC. It discusses the JLVC Interoperability Working Group and delineates the roles and responsibilities of that group. This section also addresses the information assurance guidelines.
- Section 7 establishes uniform configuration managment practices for the JLVC software development, integration, and delivery.
- The Appendices include the operational specifications of the JLVC by functional area. This part of the document details how the different functional areas work. It includes federate representation responsibilities.

2.0 Background

2.1 Purpose and Objective

The purpose of the JLVC Federation is to support more realistic and more effective joint training for joint warfighters from the tactical level through the operational level of military operations.

The objectives of the JLVC are:

- Integrate Service, Joint, and other models, simulations (M&S) and tools to stimulate and support joint training.
- Stimulate Service, Joint, and Multi-national C4I systems with reasonable, complete, and consistent M&S-generated data.
- Support Service training of Joint Tactical Tasks.

2.2 Live, Virtual, and Constructive Training

Now, and even more importantly in the future, commanders will use the joint live, virtual, and constructive (L-V-C) environment to train all units of a particular organization, combined arms team, or Joint Task Force (JTF) simultaneously. Assembling all units of a task-organized force involved in live training at the same time and place is becoming increasingly difficult. Recognizing the numerous training options and joint L-V-C training resources available are prerequisites to planning and conducting cost-effective training.

Understanding how to conduct tough, realistic training at every tier sets the foundation for successful multi-echelon, joint, interagency, and coalition operations. To train for these types of operations, Commanders' use a mix of joint L-V-C training to achieve and sustain command, unit and staff proficiency on selected core METL (CMETL) and directed METL tasks and supporting unit and staff battle tasks. The goal is to train mission essential tasks to standard and sustain a wartime readiness posture. Combatant Commands, Joint Task Forces and Service Component organizations rely more on virtual-constructive (V/C) training events to attain and sustain warfighting proficiency. Commanders at the Service Tactical and Unit and Crew levels attain and sustain warfighting proficiency and develop warrior field-craft primarily through virtual-live (V/L) training.

2.2.1 JLVC Training Mix

Table 1 provides one possible set of options available to commanders to train individuals, staffs, leaders, units, and themselves using a mix of joint L-V-C events to support crawl-walk-run training. The commander selects the tools that will result in the unit receiving the best training based on available resources. V-C training events cannot replace live training. They can, however, supplement, enhance, and complement live training to sustain unit proficiency.

Table 1. Joint Live, Virtual, and Constructive Training Mix

Several Options: Commanders Select the Mix									
Tier	**Leaders**			**Staffs**			**Units**		
	Crawl	**Walk**	**Run**	**Crawl**	**Walk**	**Run**	**Crawl**	**Walk**	**Run**
1 – Combatant Command	C	V/C	L/V/C	C	V/C	L/V/C	C	V/C	L/V/C
2 – Joint Task Force	C	V/C	L/V/C	C	V/C	L/V/C	C	V/C	L/V/C
3 – Service Components	C	V/C	L/V/C	C	V/C	L/V/C	C	V/C	L/V/C
4 – Service Tactical	V/C	L/V	L	V/C	L/V	L	V/C	L/V	L
5 – Unit and Crew	V/C	L/V	L	V/C	L/V	L	V/C	L/V	L

Notes:

- *Live (L): Real people operating real systems to include both live people operating real platforms or systems on a training range and battle staffs from joint, component or service tactical headquarters using real world command and control systems.*
- *Virtual (V): Real people operating simulated systems. Virtual simulations inject humans-in-the-loop in a central role by exercising motor control skills (e.g., flying an airplane), decision skills (e.g., clearing fires), or communication skills (e.g., as members of a C4I team).*
- *Constructive (C): Models and simulations that involve simulated people operating simulated systems. Real people make inputs to such simulations, but are not involved in determining the outcomes.*

2.2.2 JLVC Support of Tier Training

Technology provides the real-time capability to link multi-echelon, joint, interagency, and coalition elements into a comprehensive training environment. A unit may potentially conduct an exercise with elements training in L-V-C simultaneously using semi-automated forces and simulations. With improvements in the facilities available for training, units can realistically train for new and more complex missions. Commanders can tailor a variety of joint L-V-C training tools to support various missions and local circumstances and/or to take advantage of networked systems to enhance training and rehearse missions. By leveraging technology and information systems, commanders minimize role player and support requirements, and maximize the training of as many leaders, staffs, units, and warfighters as possible.

2.3 Use Cases

The first use of a JLVC-like environment was the experiment Millennium Challenge 2002 (MC02). MC02 was a major war game experiment conducted by the United

States armed forces in mid-2002 and was likely the largest such event in history. It was meant to be a test of future military "transformation"—a transition toward new technologies that would enable network-centric warfare and provide more powerful weaponry and tactics. The event, which ran from July 24 to August 15 at considerable cost, involved both live forces at nine different training sites and computer simulations at seventeen different locations. MC02 opened the door to a new paradigm in training, where simulations would fill the gaps in training when forces were either not available to participate in large-scale live training events or the costs were prohibitive. This shift also resulted in a requirements-based training philosophy that tailored training to a command's needs, not a schedule. These efforts were specifically geared to more effectively train Department of Defense (DoD) forces, better utilize training resources, and reduce the impact on personnel and equipment. This holistic view of training led to the Joint National Training Capability (JNTC)-developed JLVC training environment. To this end, the JNTC development team has developed an architecture that allows Joint and Service simulations to be integrated in a tailorable, scalable and cost effective fashion to meet specific training goals and objectives.

United States Joint Forces Command (USJFCOM) J7 has used the JLVC Federation for almost five years to support both service and combatant command (COCOM) exercises. Initially JLVC use was restricted to specific JNTC events that were designed to showcase its ability to combine the live, virtual, and constructive training environments. Major exercises that the JLVC Federation has supported include:

Table 2. Major Exercises Supported by JLVC Federation

Exercise	Exercise Sponsor
Western Range Complex 04 (WRC 04)	JNTC, Services
Joint Readiness Training Center – Air Warrior II (JRTC-AWII) 04	JNTC, Services
Commander Joint Task Force Exercise 04 (CJTFEX 04)	JNTC, Services
Joint Red Flag 05	JNTC, Services
Unified Endeavor 05-2, 06-1, 07-1, 08-1 (UE)	CENTCOM
Terminal Fury/Global Lightning 07, Terminal Fury/Turbo Distribution 08, Terminal Fury 09 (TF), Talisman Saber 07, 09 (TS)	PACOM, STRATCOM, TRANSCOM, Australia
Austere Challenge 06, 08 (AC)	EUCOM
Ardent Sentry – Northern Edge 07 (AS/NE 07)	NORAD-NORTHCOM

One of the goals of the JNTC program was to link the various major service training sites within the continental United States (CONUS) (and eventually OCONUS) so that live, virtual, and constructive forces could be brought together virtually, thereby improving training while realizing resource savings. The early JNTC events successfully proved this concept, allowing forces from different services and at widely dispersed locations to train together without having to physically co-locate. In order to facilitate this goal the JLVC Federation had to be able to accomplish the following:

- Provide realistic stimulation of joint and service command and control systems.
- Allow each service to utilize its primary service constructive simulation within a federation that includes other service constructive simulations, thus creating a joint constructive environment in which to replicate joint combat operations to a high level of detail.
- Promote the interoperability of instrumentation equipment (Multiple Integrated Laser Engagement System (MILES) gear, aircraft pods), virtual simulators, and constructive simulations in order to allow all training environments to train in a common synthetic battlespace.
- Support federation interoperability over a distributed architecture, so that elements of that federation need not be co-located in order to properly function.
- Initially, no exact requirements were levied concerning scalability, although throughout its development, the JLVC Federation has consistently developed the capability to handle larger scenarios that take place within larger geographic areas.

During the FY 06 time period, the JLVC Federation began to see use in COCOM training events – for two primary reasons.

- To take advantage of specialized simulations that could integrate into a larger synthetic environment (i.e. National Wargaming System (NWARS) use to simulate satellite surveillance).
- To train Tiers 3 through 5 during combatant command events. A prime example of this is the Terminal Fury exercise series in which JTF components participate, requiring greater fidelity in tactical level operations. In particular, a very high level of fidelity was required in air and maritime operations, along with detailed representation of intelligence, force deployment, missile defense, and Common Operational Picture (COP) fusion and management.

Use of the JLVC Federation to support COCOM events has focused much more heavily on the constructive training environment, with some virtual simulator use, but almost no integration of live instrumented forces (the exception has been Talisman Saber). COCOM events have required much larger entity counts and the ability to represent very large geographic regions, especially for those events, which have included multiple COCOM Areas of Responsibility (AORs) (i.e. TF/GL 07).

With its roots at the tactical level, the JLVC Federation has proven its usefulness to train all training audience tiers (1-5). Its use of service and agency training simulations helps to ensure that operations in the various domains are represented accurately. The primary drawbacks to the use of the JLVC Federation are that it can (depending on the scenario and training audience) require a relatively high level of resources to run (simulation operators, equipment) and that the architecture can become quite complex.

2.4 Conceptual Model

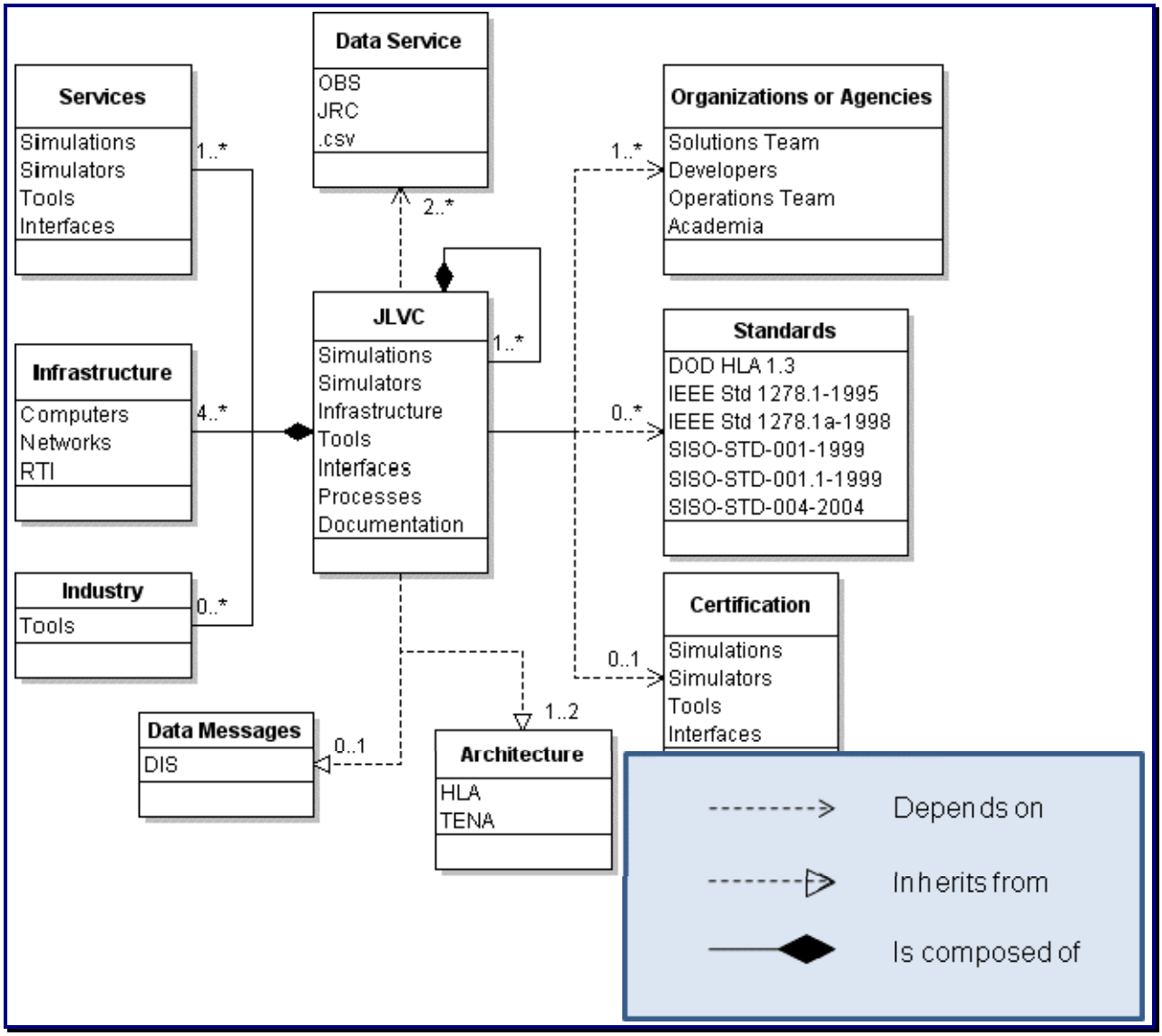

Figure 1. JLVC Conceptual Model

2.5 High-Level Requirements

Listed below are the high-level requirements for a joint LVC training environment.

- Use Service simulations and tools nominated and certified by Joint Warfighting Center (JWFC) in a cost effective and efficient manner to accomplish the training objectives.

- Represent individual people, vehicles, aircraft, and vessels explicitly with sufficient granularity:
  - To enable acquisition by Signals Intelligence (SIGINT), Imagery Intelligence (IMINT), and Human Intelligence (HUMINT) sensors (Intel)
  - To flow forces and consume supplies down to the individual item level of issue (Logistics)
  - To interact with the environment and among each other with the appropriate level of fidelity (Firepower, Maneuver)
  - To be able to be accurately represented on a COP
- Display flexibility in reacting to changing exercise demands (Federation Management; Scenario Generation).
- Stimulate all Global Command and Control Station (GCCS) mission areas (operations, mobilization, deployment, employment, sustainment, and intelligence) and Service C4I systems including:
  - Command and Control Personal Computer (C2PC), Air Defense Operations Center System (ADOCS), Theater Battle Management Core System (TBMCS), Air Defense Systems Integrator (ADSI), Internet Operating System (IOSv1), AFATDS, Air and Missile Defense Workstations (AMDWS), and Area Air Defense Commander (AADC)
  - Generic Area Limitation Environment Lite (GALE LITE) and All Source Analysis System (ASAS) LITE
  - ISR-M and Joint Services Work Station (JSWS)
- Enable real-time (1:1) execution.
- Enable composability based on training audience requirements, i.e. federates, systems, or tools may change depending on training audience requirements.

3.0 Overview

3.1 Definitions of LVC

The JLVC Federation is an integrated training capability that generates a realistic simulation of the real world in order to facilitate a live training audience's accomplishment of joint training objectives. It provides an integrated and coherent LVC training environment that includes appropriate joint context, and allows global training and mission rehearsal in support of specific joint requirements. The domains of the LVC environment are defined as follows:

- Live: Real people operating real systems to include both live people operating real platforms or systems on a training range and battle staffs from joint, component or service tactical headquarters using real world command and control systems.
- Virtual Simulation: Real people operating simulated systems. Virtual simulations inject humans-in-the-loop in a central role by exercising motor control skills (e.g., flying an airplane), decision skills (e.g., clearing fires), or communication skills (e.g., as members of a C4I team).
- Constructive Model or Simulation: Models and simulations that involve simulated people operating simulated systems. Real people make inputs to such simulations, but are not involved in determining the outcomes.

At its core, the JLVC Federation consists of a set of simulation applications that use a mixed architecture infrastructure to distribute simulation data globally across wide area networks and interface devices that allow simulation data to interface with live systems and embedded training devices on live platforms. The following components comprise the JLVC Federation:

- M&S software (federates).
- Distributed simulation infrastructure, e.g. the Run Time Infrastructure, federation, management and gateway software, Distributed Interactive Simulation (DIS) enumeration loggers/checkers, etc.
- Air, land, sea, and undersea tactical training ranges.
- Training systems embedded in combat systems to support training the combat system, e.g. Battle Force Tactical Trainer support for onboard ship training, Patriot training systems for Patriot system training, etc.
- Stimulated command and control systems (usually Global Command and Control System-Joint (GCCS-J)).
- The Joint Training and Experimentation Network (JTEN).
- C4I interfaces.

The JLVC architecture supports training at the operational staff level down to the tactical level. As technology changes, the basic JLVC architecture concept remains the same; no central computer controls the entire simulation exercise. As new host computers are added to the network, each brings its own resources.

Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities. Simulations may also maintain a model of the state of the environment and non-dynamic entities, such as bridges and buildings, which may be intact or destroyed.

3.2 Federation Description

3.2.1 Federates and Functional Allocation

The primary systems comprising JLVC and their functional allocation are included below. Other systems have previously been included in JLVC Federation execution and will be included again in the future; this is not an all-inclusive list:

- Joint Conflict and Tactical Simulation (JCATS); ground and Special Operations Forces (SOF)
- JCATS Low Overhead Driver (JLOD); “wrap around” ground
- Air Warfare Simulation (AWSIM); air
- Joint Semi-Automated Forces (JSAF); maritime
- Air & Space Constructive Environment Information Operations Simulation (ACE-IOS); intelligence
- Tactical Simulation (TACSIM); intelligence
- National Wargaming Simulation Next Generation (NWARS NG); intelligence
- Joint Deployment Logistics Module (JDLM); logistics

3.2.2 Federation Data Services Design

Data services for the JLVC Federation were designed to redress several issues that plagued JWFC in exercise preparation. Principle among these issues was the need for correlated data among the systems comprising an instance of the federation execution. En route to designing a means of producing correlated data, the data services design addressed additional issues as follows:

- Source data issues. Source data comes from a variety of sources and is often incomplete or is difficult and time consuming to get. Source data can contain inaccuracies and different authoritative sources provide data, which can be inconsistent. Source data issues dictated developing a repository from which data can be drawn when needed. The repository currently includes order of

battle and terrain data. Additional data types, for example, target data, are being added as the repository matures.

- Data ownership and accessibility. A key component of JWFC's approach to exercise support is the training audience's responsibility to take ownership of the data. Ownership is formalized in the certification of the data by the exercise director. Data ownership is imperative since the training audience typically knows far more than those at JWFC about the actual order of battle, target lists, and other data-sensitive components. This dictates that the training audience be given time and access to the data for pre-exercise data review. This in turn led to development of a web-based capability with appropriate permissions and access restrictions and a process involving JWFC posting initial data sets for subsequent training audience review and change.
- Correlated data. The importance of correlated data has already been mentioned. In the absence of correlated data, disparate systems would present an inconsistent picture to the training audience. The importance of correlated data led the design team to develop a process and tools to create a single exercise-ready data set, and then provide data from the set to the disparate systems comprising the federation execution in a standard, extensible data format, eXtensible Markup Language (XML). The single data set would necessarily contain a super-set of all data required by all systems and each system would be responsible for importing and using only the data it required.
- Timeliness and flexibility. Clearly data preparation should be complete in sufficient time prior to the exercise and pre-exercise events to allow training audience review and certification. Yet system flexibility is important to support "last minute" changes in unit STARTEX locations, etc. Moreover, the "objective" system would be able to support data generation for Mission Rehearsal purposes, necessitating data preparation in hours. Timeliness and flexibility underscored the necessity of a web-based approach and imposed performance requirements toward which the development team is still working.
- Usability. The role played by training audience representatives in reviewing and changing data imposed requirements for user-friendly interfaces.

In summary, the data services design incorporates a data repository from which data for an exercise is pulled, web-services are employed to present the data to the training audience for review, correction, and certification forming a single exercise-ready data set. This data set is then provided to the disparate systems comprising the federation execution with each system being responsible for importing and using the data it required.

4.0 Architecture

4.1 Live Environment

To begin the discussion of JLVC architecture, it is important to remember that the JLVC Federation exists only to improve joint training for the joint warfighter. The term "live environment" encompasses the full range of soldiers, sailors, airmen and Marines – including commanders and staffs - who make up the live training audience. It also includes all the military platforms, vehicles, weapons and information systems that this training audience employs. These live warfighters are the central focus of the JLVC architecture.

To support the accomplishment of joint training tasks, joint warfighters must perceive the world during training in the same way that they do during real world operations. Otherwise, ineffective or even negative training can occur. Thus, stimulation of the live environment represents the ultimate purpose for the JLVC Federation's simulated representation of the environment. It is only through live sensors and systems that the training audience perceives the world and executes actions in the battlespace.

Because the JLVC Federation supports training at both the operational and tactical levels, it must stimulate the live environment at both. At the operational level, the training audience is typically located in a headquarters facility or command center. At this level, the training audience primarily perceives the world through C4I systems. Therefore, delivery of training capability to an operational training audience is primarily accomplished by injecting simulation data into live C4I systems. These C4I systems encompass the full range of military information systems available to a commander and staff, to include command and control systems that support maintaining situational awareness, tactical data links for exchanging real-time information, intelligence collection systems and databases, fires planning and execution tools, decision support tools, logistics and transportation systems and collaborative tools.

Since joint training at the tactical level usually occurs at various live ranges, training facilities or simulators that are owned and operated by the Services, the JLVC Federation architecture provides the means to exchange data with service systems to stimulate those ranges and facilities. The methods for exchanging data with Service sites will be discussed below, but how the Services actually stimulate live platforms, sensor and weapons systems is beyond the scope of this document[1].

[1] In fact, the ability to directly stimulate tactical sensors and weapons systems with simulated data is one of the most fundamental gaps in current joint training capabilities. Until real-world platforms and systems fully incorporate embedded training capabilities, which allow direct stimulation of live sensor and weapons systems from simulated inputs, this gap will persist.

4.2 The JLVC Technical Architecture

The JLVC technical architecture provides interoperability between live, virtual and constructive systems employed in the joint training environment using the following primary architectures, protocols and messaging standards:

1. High Level Architecture (HLA) - High-Level Architecture Interface Specification, Version 1.3, U.S. Department of Defense, 2 April 1998.
2. Distributed Interactive Simulation (DIS) – IEEE Standard 1278.1a-1998, IEEE Standard for Distributed Interactive Simulation.
3. Test and Training Enabling Architecture (TENA) – TENA Architecture Reference Document, version 2002.
4. Over-the-Horizon Targeting GOLD – Operational Specification for Over-the-Horizon Targeting GOLD (OS-OTG), Baseline 2004
5. Link 16 (Link 16) – MIL STD 6016C, DoD Interface Standard – Tactical Data Link (TDL) 16 Message Standard
6. United States Message Text Format (USMTF) – MIL STD 6040, DoD Interface Standard – U.S. Message Test Formatting Program, Baseline 2008.

These architectures, protocols and messaging standards are the primary means by which the JLVC Federation is able to facilitate the accomplishment of joint training objectives.

4.2.1 JLVC Simulation Architecture

As described in section 2.2.1, the JLVC Federation supports different training audiences with a variety of L-V-C components. This flexibility requires a “composable” simulation architecture enabling selection of systems based on exercise requirements. As such, there is no single or static representation of the JLVC simulation architecture. **Error! Reference source not found.** displays an example architecture that includes the core components described in section 3.2.1.

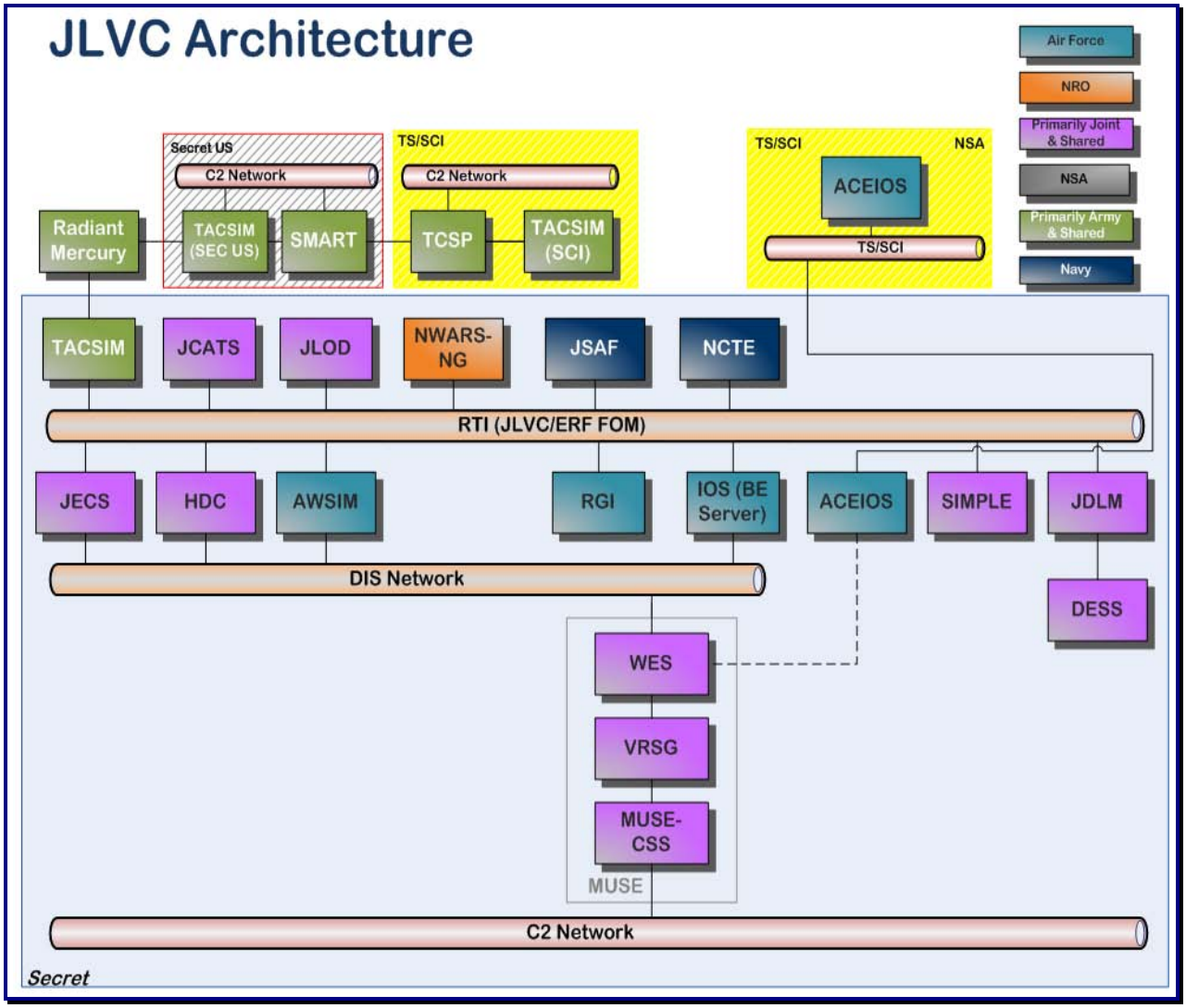

Figure 2. The JLVC Federation Architecture

4.2.1.1 High Level Architecture

HLA data formats are specified in a file format called the Object Model Template (OMT). The FOM is the version of the OMT used to specify interoperability standards between federates. The HLA RTI optimizes network bandwidth by using a dynamic allocation of multicast channels to limit updates to only those federates that have subscribed to that type of update. Local reflection of entities also avoids retransmitting any attributes that have not changed since the last update.

The primary mechanism for intra-federation communications will be an RTI implementing the HLA Interface Specification version 1.3 using RTI Initialization Data (RID) parameters added in support of MC02/ Distributed Continuous Experimentation Environment (DCEE)/JNTC. The JLVC currently uses the Raytheon - Virtual Technology Corporation (VTC) RTI NGPro version 4.0.4. The laboratory version is VTC NGPro 4.2.4.

4.2.1.2 Distributed Interactive Simulation Protocol

The DIS protocol is an older protocol for distributed simulation that was to be replaced by HLA. However, DIS is still used by many simulation applications throughout DoD. DIS is a message-based system where information is broadcast across the wide area network each time one attribute of that message type changes. Proxy copies of remote entities are still maintained on each simulation.

JLVC's HLA-DIS gateways allow the federation to operate using the HLA specification. Legacy systems relying on DIS may only integrate with JLVC upon approval from the JLVC Federation Manager. For the DIS Site Identifiers and Applications Identifiers, refer to section 5.1.7.

4.2.2 JLVC C4I Architecture

4.2.2.1 Live Environment Stimulation at the Operational Level

To stimulate the operational training audience's C4I systems, the JLVC Federation employs a number of interface devices. These devices provide gateway services between simulated data in the virtual/constructive domain and live C4I systems. For an operational level training audience, the primary system to be stimulated is the Global Command and Control System (GCCS), which is the primary system that supports development of the COP.

4.2.2.2 Common Operational Picture

The COP is a near real time situational display of friendly, neutral and enemy ground, maritime and air units that includes relevant graphic overlays, intelligence products and tactical decision aids. While the term "COP" is often synonymous with GCCS, the COP actually refers to the overall set of joint, Service and multinational partner C4I systems that combine to generate, display and disseminate one integrated database of objects and actions in the current operational environment. GCCS is the core C4I system that supports the COP, but there are many other systems that are involved.

The term "TOP COP" refers to the actual GCCS system that hosts the primary COP node for the appropriate theater's Combatant Commander. For USJFCOM-sponsored exercises, the TOP COP is usually located in the JWFC. The TOP COP represents the union of joint and Service common tactical pictures augmented by national and theater intelligence feeds, Air Tasking Order (ATO) data and meteorological and oceanographic data. The TOP COP provides a consolidated, centralized operational database for distribution to C4I nodes that support the live training audience during joint training events. Figure 3 provides a stylized view of the relationship of simulations, information management tools and C4I systems combining to support the live training audience during a joint training event.

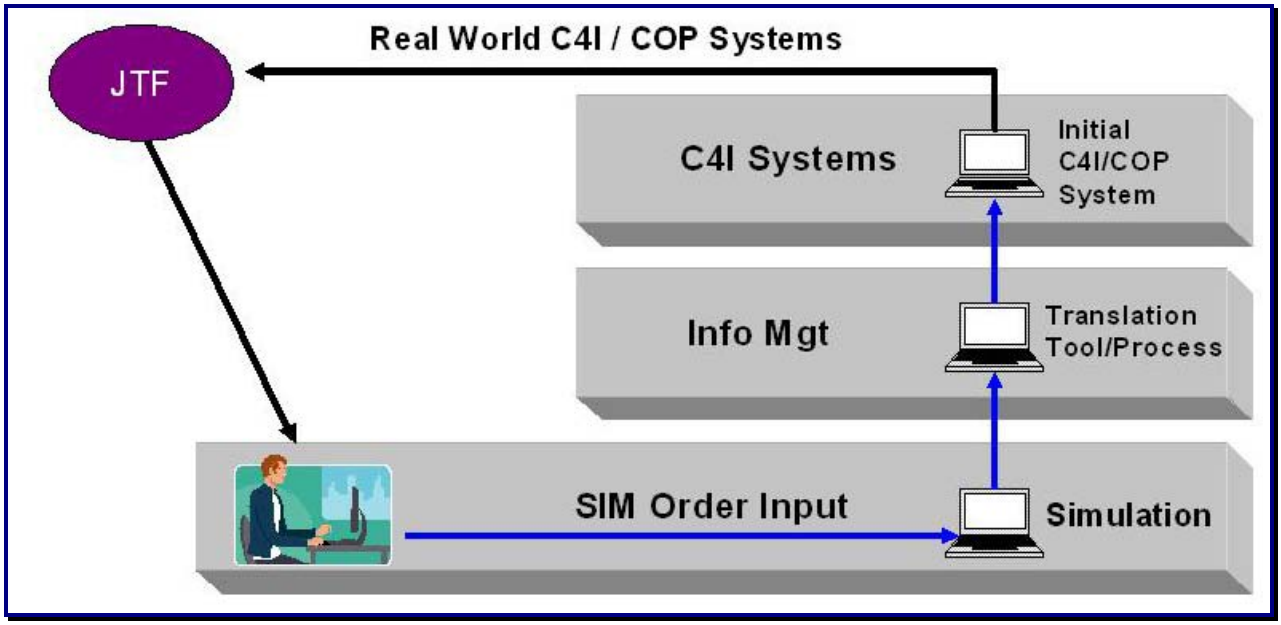

Figure 3. The Simulation Feeds the COP

4.2.2.3 C4I Stimulation

During joint training events, the JLVC Federation will produce the majority of sensor, track and intelligence inputs into the C4I architecture. Simulation data is transformed into C4I data and messages via a number of different simulation-to-C4I gateways or software bridges.

GCCS is the primary C4I system used to build and share the COP. GCCS receives inputs from a variety of sources, to include both real and simulated forces. The following paragraphs describe the interfaces to the COP during joint training events.

4.2.2.4 Tactical Digital Information Link (TADIL) J/Link 16

Link 16 is a tactical information system designed to exchange surveillance and command and control (C2) information among various C2 platforms and weapons platforms. Sometimes termed the “Joint Data Network,” Link 16 is a network of communications, navigation, identification and weapons control systems that exchange information in near real-time. Link 16 is one of the primary feeds that support Service and component generation of the common tactical picture.

Link 16 inputs during joint training events include a variety of live, virtual and constructive radar and sensor platforms that provide real-time air, space, maritime and ground tracks on contacts of interest. Much of the virtual and constructive input to Link 16 comes from simulated Link 16 reporting units, such as a virtual Patriot Flight Mission Simulator – Digital (FMS-D), an AWSIM E-3 Sentry Airborne Warning And Control System (AWACS) constructive aircraft or a JSAF guided missile destroyer (DDG).

It is the responsibility of each of these virtual and constructive systems to correctly perform the surveillance functions of reporting and updating surveillance tracks, correlating and de-correlating tracks, performing combat identification, transferring reporting responsibility (R2) and dropping surveillance tracks. Each system must ensure that:

- Link 16 J-series messages are accurate, complete, timely and compliant with MIL-STD 6016C, to include proper message sequencing.
- Track quality (TQ) is accurately modeled and reported.
- A single track is reported for a single entity based on correct modeling of R2.
- Tracks are reported with the correct track block as assigned in the Operational Tasking (OPTASK) Link.
- Precise Participant Location and Identification (PPLI) messages contain the appropriate JU as specified in the OPTASK Link.
- The simulation indicator bit is correctly set in the constructive systems' PPLI and surveillance track messages.
- PPLI messages contain appropriate navigational information.
- The simulation bit setting is typically exercise- or system-dependent. Virtual and constructive systems should be able to toggle this bit between "live" or "simulated" depending on exercise requirements, or have specialized equipment that handles changing this indicator in outgoing J messages. This ability is typically required to support certain older combat systems that do not support the simulation indicator bit.

The Joint Interface Control Officer (JICO), in coordination with USJFCOM C4I exercise planners, will specify reporting procedures for both live and virtual/constructive Link 16 participants. In most cases, virtual and constructive systems will report Link 16 data to an ADSI, or similar system, for injection in to the exercise Link 16 network. Various protocols can be used to transfer data to and from the ADSI, such as the Simulation to C4I Interchange Module for Plans, Logisitics and Exercises (SIMPLE) or Multi- TADIL Capability (MTC) protocols. SIMPLE and MTC are supported by both ADSI and GCCS, as well as many other Link 16 devices. The ADSI supports network transfer of Link 16 data, meaning that an exercise Link 16 network can connect participants from around the globe and allow a distributed training audience to carry out "localized" Link 16 operations as it would in the real world.

One common issue with virtual and constructive Link 16 producers involves a failure to perform R2, which results in dual surveillance tracks, and therefore a cluttered and inaccurate Link 16 picture. Constructive Link 16 simulations are typically designed to generate Link 16 J-series messages to support test or training events with little or no requirement for interfacing with other Link 16 producers or human-in-the-loop operators.

In this stand-alone role, constructive Link 16 models are affordable alternatives to live systems for developing a limited surveillance picture. However, because most constructive systems do not receive and process all J-series messages required to perform real-world surveillance functions, they are limited in their ability to participate in live Link 16 networks where interaction with other Link 16 systems is essential. In particular, their inability to perform R2 is a significant problem when trying to integrate constructive Link 16 systems with live or virtual systems. To resolve this problem, several systems exist that perform a proxy service (i.e., “Link Proxy”) for constructive simulations that are not fully compliant with the MIL-STD 6016C specification. This function is intended to compensate for existing limitations in constructive Link producers and allow them to interoperate in a live Link 16 network.

Besides being used for real-time surveillance and weapons control, Link 16 data is also injected into GCCS as a primary COP input. It is important to point out that there can be only one Link 16 inject point per track type in the exercise architecture. Otherwise, multiple track problems and data loops are guaranteed.

4.2.2.5 Unit Reporting

Each component in a joint exercise is tasked to generate and disseminate unit information in the COP. This data can be reported manually or automatically, and includes both live and simulated forces. Additionally, simulated sensor contacts of air, sea, and land data will be selectively integrated with the COP to support training requirements.

Live units typically follow standard procedures for reporting information into the COP. Units with GCCS-compatible terminals and connectivity normally report their data electronically, while units without GCCS capability may be tasked to report their data to a designated unit or component with GCCS-compatible terminals. The designated unit then enters the data into GCCS, either through a GCCS process directly or from an automated system with a GCCS interface.

Virtual and constructive units must follow similar procedures. Typically, each constructive simulation that provides unit reports to the COP has its own specialized interface device for doing so. For example, JCATS primarily uses the SIMPLE application, while JSAF uses the C4I Gateway or the JLVCDT. Regardless of the device, the outgoing messages must meet the OS-OTG messaging standard for the message types that they send. The most common message types used are the Contact Report (GOLD), Extended Contact Report (XCTC) and Joint Unit Report (JUNIT). Other OS-OTG messages are also used, such as Overlay (OVLY2 or OVLY3) messages, but are much less common.

In addition to self-reporting, simulated units may make reports to the COP that represent sensed units and platforms in the battlespace. These units can be other friendly units, opposing force units or neutral units or platforms of interest. In these cases, it is essential that the constructive model accurately represent sensor

performance, to include contact position and position error, report timeliness, sensor coverage, dwell time and revisit rate.

GCCS reports from simulated forces can represent either individual unit reports, as is the standard in the Navy, or “roll up” reports in which a senior commander reports status for subordinate units as well, as is common in both the Army and Marine Corps. To assist real world track management, it is important that whatever method is used by the reporting system represent as realistic a track feed as possible, and that human operators are in the loop to monitor the quality of the simulated track feed. The most important aspects in making the simulated track feed as realistic as possible are explained in the following paragraphs:

Report Periodicity

During joint training events, the COP Manager sets and enforces reporting requirements. The COP Manager adjusts these reporting requirements and track lifespan times as required to maintain an accurate and timely COP. While there is no clear rule of thumb for track reporting periodicity, virtual and constructive systems should be able to set track update rates by domain and force. GCCS track update time requirements will vary by exercise.

Track Type

In order to support track management functions, all tracks generated from virtual and constructive systems must comply with the COP Manager’s guidance for track types. Tracks of different track types cannot be correlated or associated to Link tracks or other platform or unit tracks in GCCS. In real world operations, the Track Type field is typically left blank, indicating a live, real-world track. Depending on the exercise, though, this track type field may also be set to a value shown in Table 3:

Table 3. Track Type

Track Type	Value
LIVE TRAINING TRACK	2
SIMULATED TRAINING TRACK	3
DEMAND ENTRY - LIVE NONTRAINING TRACK	4

JLVC interface devices that connect to GCCS for track generation must be able to set the track type field according to the guidance specified for the particular exercise.

Track naming conventions are usually specified prior to joint training events. In some cases, especially when an exercise COP is connected to a real world COP, the names of simulated units reported to GCCS must be changed to prevent confusion with the actual locations and actions of those units in the real world. In these cases, various methods are used, such as pre-fixing the name of the simulated unit with an “X.”

Naming conventions will be established during the exercise planning phase by USJFCOM C4I planners.

During joint training events there are usually two separate COPs in use, one operational COP that is used by the training audience and a “ground truth COP” used by exercise control staff. In this discussion of the C4I architecture, we have so far been primarily concerned with the operational COP. However, it is important that a separate COP architecture supports exercise control personnel.

Exercise control and management requires an accurate, ground truth display of simulated and real-world units and platforms in the exercise. Therefore, during joint training events, the JLVC Federation enables a Ground Truth COP that consists of air, ground and maritime live range traffic as well as simulation ground truth. This Ground Truth COP is primarily used by the Joint Exercise Control Group (JECG). The Ground Truth COP is a stand-alone COP display component for the JECG’s use only and is completely separate from the operational COP that the training audience uses.

4.3 Test and Training Enabling Architecture (TENA)

TENA applications use the middleware and Logical Range Object Model (LROM) to transport TENA data across the network. The JNTC LROM supports interoperability between TENA assets in JLVC applications.

The TENA Middleware combines distributed shared memory, publish-subscribe, and model-driven distributed object-oriented programming paradigms into a single distributed middleware system. This allows users to rapidly develop complex reliable distributed applications.

Gateways provide a method for data from TENA applications to be used to stimulate other systems in the federation. Gateway of TENA/HLA (GOTH) is currently used to provide a gateway between TENA and HLA applications.

4.4 Joint Training Data Services (JTDS)

JTDS provides web-based scenario generation services developed to support the needs of the US DoD M&S Training Community. The JTDS objective is to evolve current capabilities to enable short notice support of Mission Rehearsal.

JTDS saves time and money producing correlated databases used by simulations and federations to support training events. JTDS comprises three services:

- Order of Battle
- Terrain
- Weather Effects

Since JTDS is web-based, it enables access for users at remote sites, thereby:

- Promoting data sharing
- Simplifying database development through user-friendly graphical user interfaces (GUIs)
- Providing access to a comprehensive data architecture process and tools
- Limiting redundant work across the department

Moreover, JTDS improves the quality of data used to support events by providing:

- Best available source data
- Correlated data from a variety of sources

4.4.1 Order of Battle Service (OBS)

OBS provides the following capabilities:

- Distributed editing and validation
  - User assignable permissions at any level
  - Data review tracking features
- Web-based data repository
  - Source data validated by Service authority
  - Merged for completeness
  - Correlated for federation use
- Subset of repository data forms exercise scenario
  - “Drag and drop” speeds selection from repository
  - Data available for review throughout exercise preparation phases
- Scenario file is generated in intermediate format (XML)
  - Federates take what they need
  - Ensures correlation
  - Saves time
  - Supports last minute changes or requirements

The OBS Architecture appears below as Figure 4.

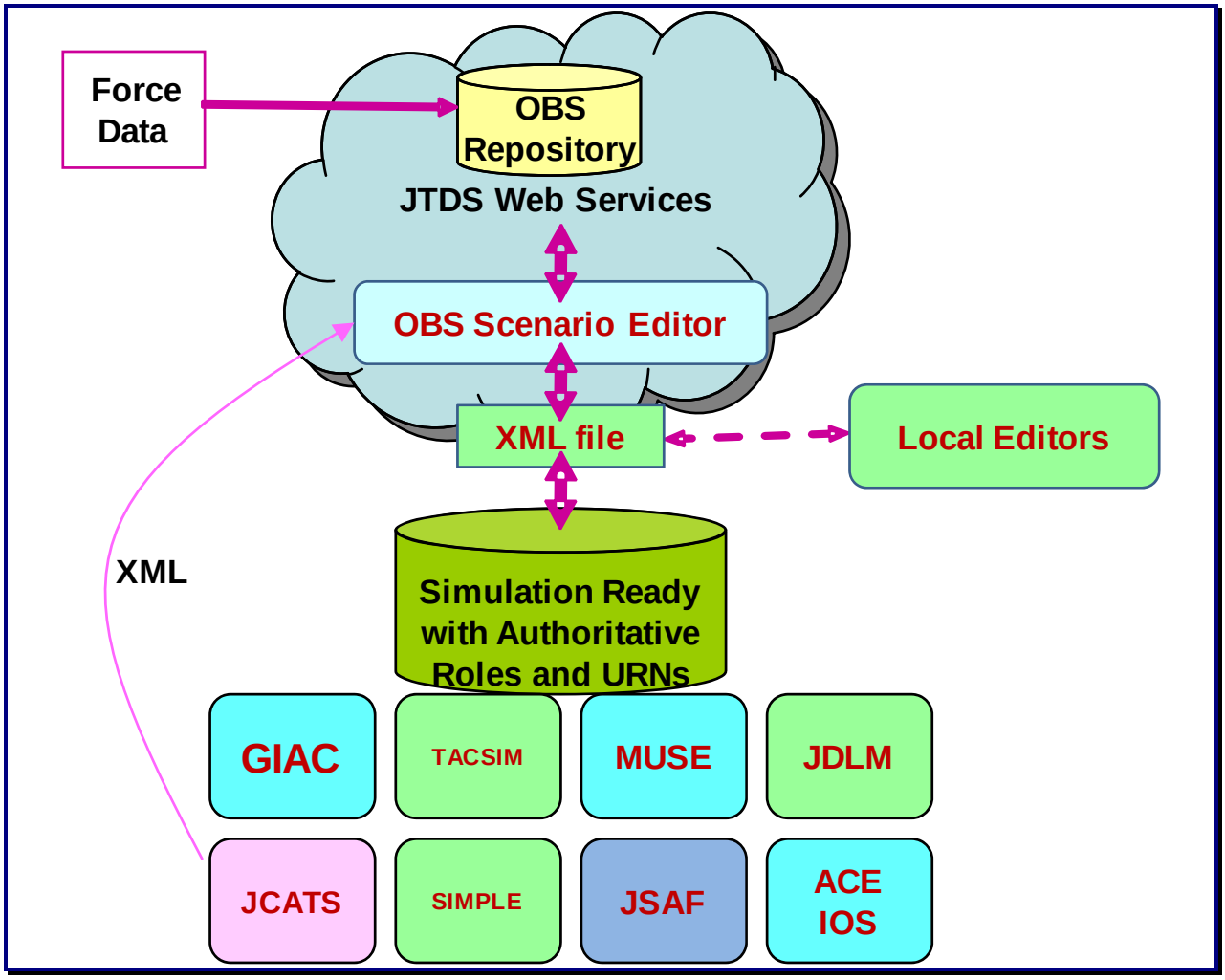

Figure 4. OBS Architecture

The current OBS schema (v1.4) is included in 7.5.2APPENDIX D:.

4.4.2 Terrain Generation Service (TGS)

TGS imports many standard authoritative source data types, and exports JCATS and Joint Theater Level Simulation (JTLS) formatted databases. The automated production process usually takes a few hours depending on the number of build requests queued and database complexity. When the database is completed, the customer receives email notification and can retrieve the terrain file from the JTDS Portal where it will be maintained with appropriate metadata for discovery and reuse by future customers. The TGS architecture appears as Figure 5.

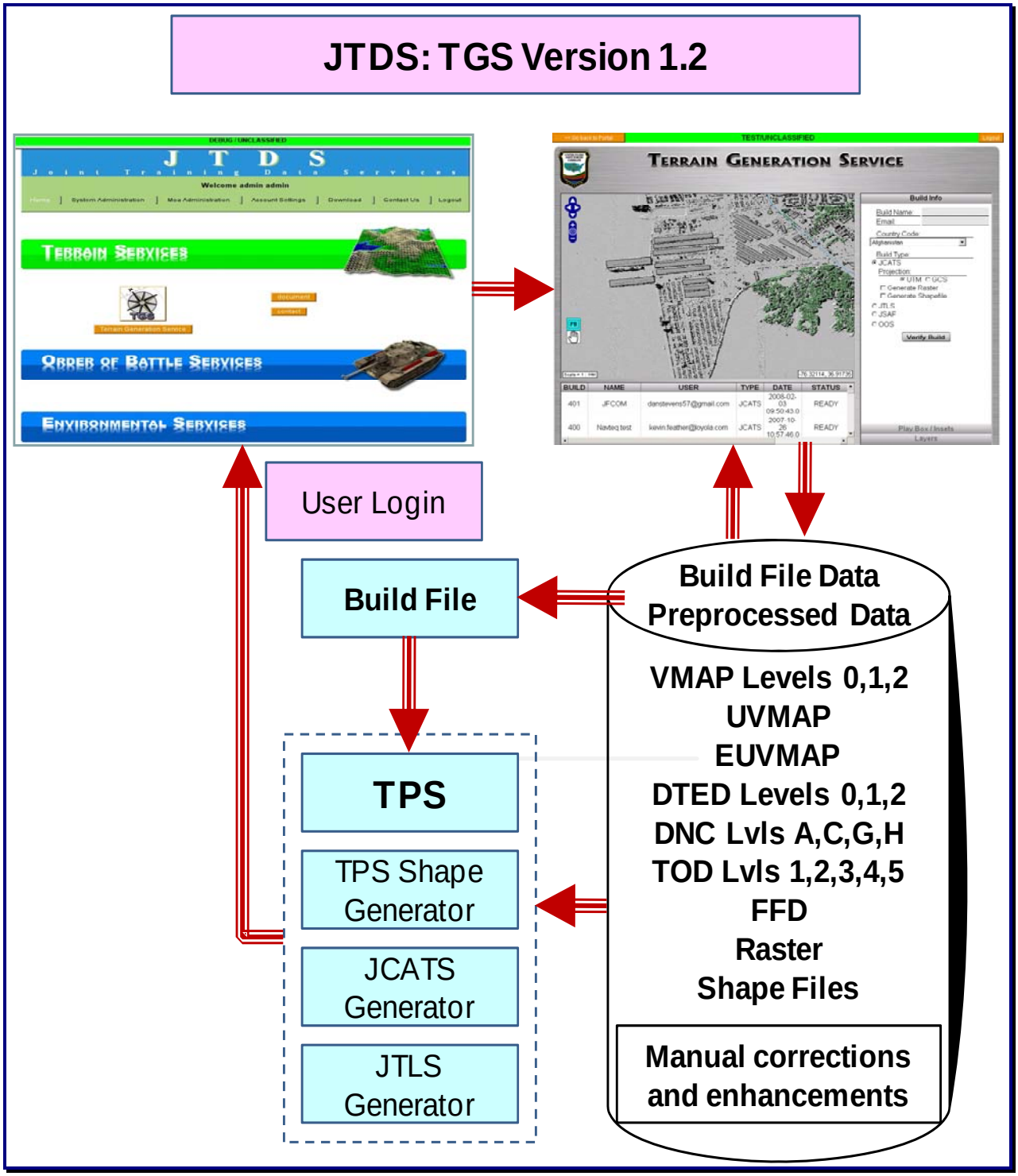

Figure 5. TGS Version 1.2

4.4.3 Weather Service

4.4.3.1 Overview

The Environmental Data Cube Support System (EDCSS) generates and provides a consistent environmental scenario to JLVC systems. To produce a scenario with desired effects, EDCSS leverages the Environmental Scenario Generator (ESG) to search historical archives or run operational environmental models with prescribed conditions. EDCSS builds upon the ESG scenario and produces environmental products tailored to meet the input requirements of each JLVC system. As illustrated in Figure 6, the EDCSS Distributor makes these products available in one of two ways: a) via coordination between the EDCSS Distributor and the JLVC-DT to the HLA JLVC FOM, or b) via a simple run-time client that requests and receives products directly from the Distributor. EDCSS ensures all systems have a consistent view of the environment and delivers products in formats required by each system. Weather products for each system are detailed in the following sections.

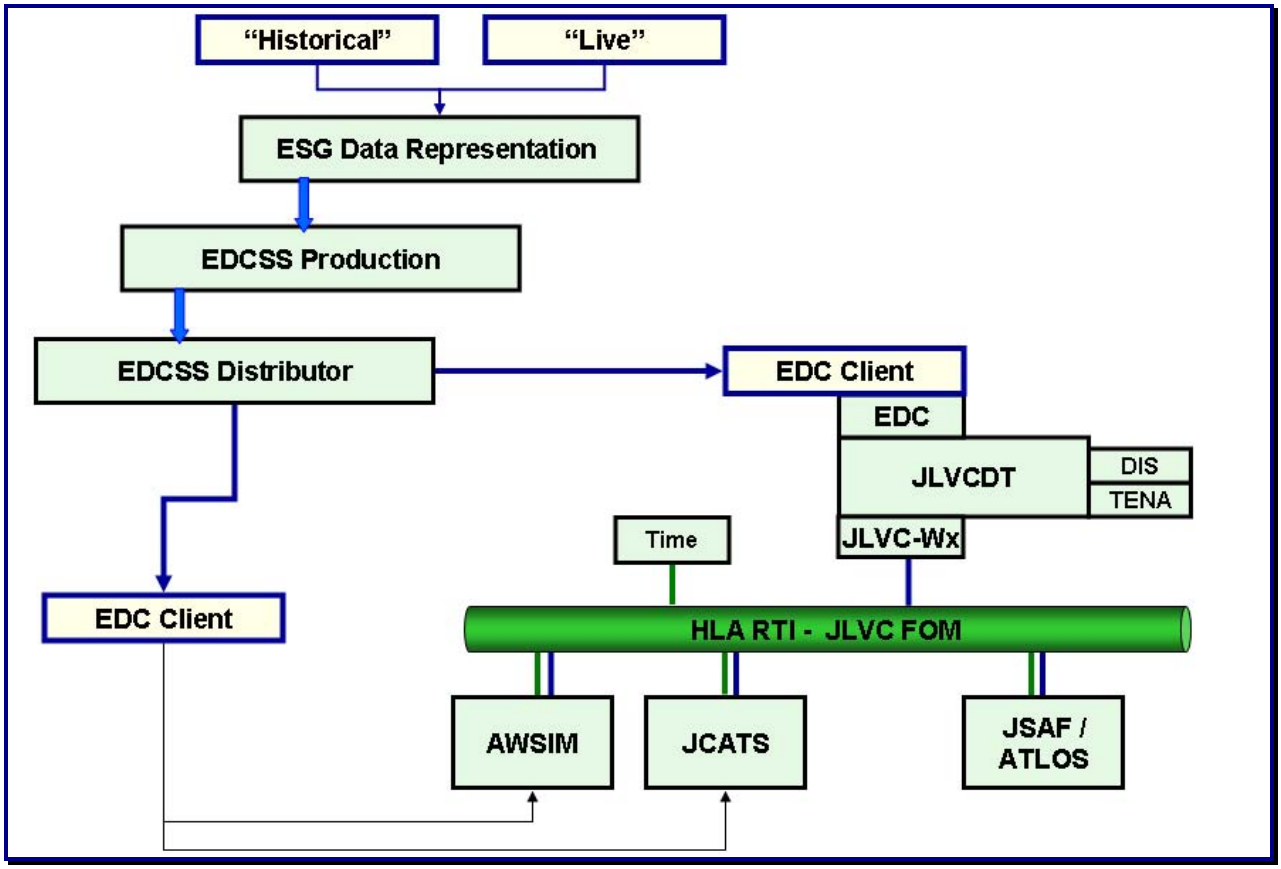

Figure 6. EDCSS Data Distribution Flow

The EDCSS footprint is small and non-intrusive. The Distributor will reside on a laptop within the JLVC test facility. Forward-deployed components manage the distribution and insertion of products into runtime simulation components. A hardware/software

system hosted at an appropriate DoD facility supports production of the required EDCSS products.

4.4.3.2 Requesting Data to Support a JLVC Event

The EDCSS Production Site provides an interface for requesting environmental data to support a JLVC event. The site is designed such that an end user must specify only high-level information, such as the area of interest, timeframe, and participating simulations for an event. Given that information, data requirements and format specifics are determined by the system, and the support process is initiated. Figure 7 shows an example from the EDCSS Production Site.

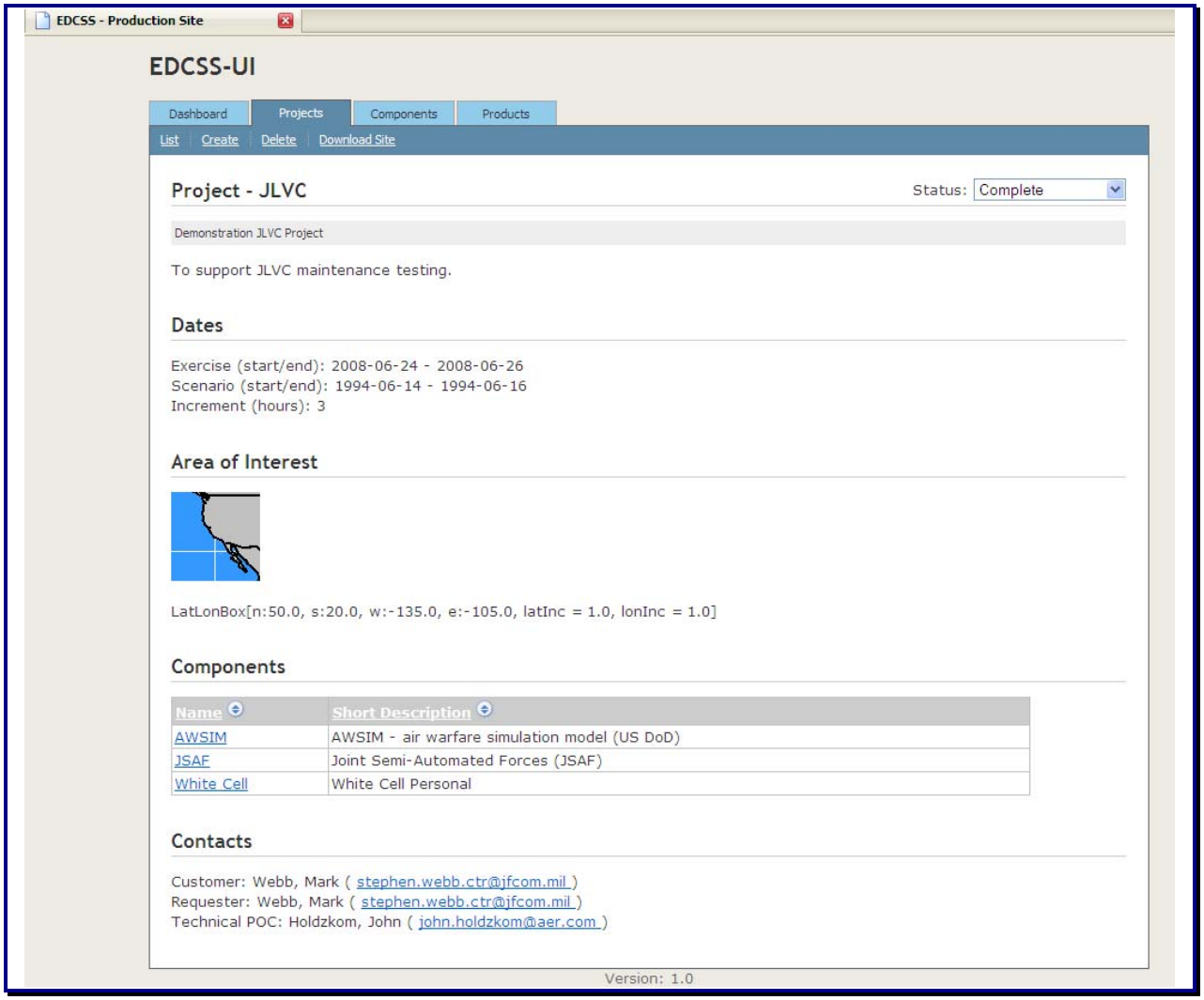

Figure 7. The EDCSS Production Site

After the products to support the participating simulations are generated, they are moved to the EDCSS Distributor from where they may be disseminated through one of the mechanisms described above.

4.4.3.3 Simulation Use of Weather

- AWSIM. AWSIM is able to ingest Hypercube and/or Weather Comma Separated Value (CSV) files. Currently this is a manual process executed at six-hour intervals. EDCSS supports this via a run-time client to periodically request and receive weather products from the EDCSS Distributor. Files are then loaded manually into AWSIM. In the future AWSIM may wish to utilize weather data published to the HLA Federate via the EDCSS–JLVCDT connection. Without Hypercube or CSV files, AWSIM has no default weather.
  - Hypercube

    The Environmental Hypercube provides pre-computed sensor performance metrics consistent with the spatial and temporal variations in weather as depicted in the Environmental Scenario Generator-produced weather scenario data set. It provides a probability of detection for a number of target types from an array of ranges and approach angles.

    The primary consumer of the Hypercube is AWSIM itself, and directions for its use are provided with the AWSIM documentation (V2.9.1.15 or later).

  - Weather CSV files

    Weather CSV files provide general meteorological parameters in a comma separated file format. Each csv file provides data for one six-hour period. Table 4 lists the provided parameters.

Table 4. Weather CSV File Parameters

Parameter	Unit	Level
Blowing Sand	boolean	Surface
Blowing Snow	boolean	Surface
Cloud Ceiling	m	Surface
Fog	%	Surface
High Turbulence Intensity	enumeration	Surface
Illumination	millilux	Surface
Low Turbulence Intensity	enumeration	Surface
Mid Turbulence Intensity	enumeration	Surface
Precipitation Intensity	enumeration	Surface
Precipitation Type	enumeration	Surface
Pressure reduced to mean sea level	Pa	Surface

Table 4. Weather CSV File Parameters

Parameter	Unit	Level
Relative Humidity	%	Surface
Temperature	K	Surface
Temperature gradient	K/m	Surface
Terrain Height	m	Surface
Thunderstorm	%	Surface
Total Cloud Cover	%	Surface
Total Precipitation	mm H2O	Surface
Visibility	m	Surface
Wind Direction	degrees true	Surface
Wind Speed	m/s	Surface
Present Weather	string	Surface

- JCATS
  - JCATS is able to ingest Weather CSV files. Currently this is a manual process typically executed at six-hour intervals. EDCSS supports this via a run-time client to periodically request and receive weather products from the EDCSS Distributor. Files are then loaded manually into JCATS. In the future JCATS may wish to utilize weather data published to the HLA Federate via the EDCSS–JLVCDT connection. JCATS has no default weather.
  - Weather CSV files provide general meteorological parameters in a comma separated file format. Each csv file provides data for one six-hour period. Table 4 lists the provided parameters.

- JSAF
  - JSAF subscribes to weather data published by EDCSS. Alternatively, it may independently set weather to default conditions; this is obviously not ideal because no other simulations in the JLVC would be playing the same weather. JSAF subscribes to weather products published by EDCSS so all federates have a consistent view of the environment.
  - JSAF ATMOSPHERE. Data is published in accordance with the JLVC FOM to meet JSAF atmospheric data requirements. Data is provided for each six-hour period. Table 5 lists the provided parameters.

Table 5. JSAF Atmospheric Parameter Requirements

Parameter	Unit	Level
Cloud Ceiling (CldCeil)	m	Surface
Precipitation Rate (PrecipRt)	kg/m/m/s	Surface
Pressure reduced to mean sea level (PresMSL)	Pa	Surface
Relative Humidity (RelHumid)	%	Surface
Temperature (Temp)	K	Surface
Total Cloud Cover (TotCldCvr)	%	Surface
Wind Speed (WndSpd)	m/s	Surface
Wind U Component (WndUComp)	m/s	Surface
Wind V Component (WndVComp)	m/s	Surface

– JSAF OCEAN 2D. Data is published in accordance with the JLVC FOM to meet JSAF two-dimensional ocean data requirements. Data is provided for each six-hour period. Table 6 lists the provided parameters.

Table 6. JSAF Ocean 2D Data Requirements

Parameter	Unit	Level
Significant Wave Height (SgWvHgt)	m	Surface

– JSAF OCEAN 3D. Data is published in accordance with the JLVC FOM to meet JSAF three-dimensional ocean data requirements. Data is provided for each six-hour period. Table 7 lists the provided parameters.

Table 7. JSAF Ocean 3D Data Requirements

Parameter	Unit	Level (meters)
U Component of Current (CurU)	m/s	0.0, 2.5, 7.5, 12.5, 17.5, 25.0, 32.5, 40.0, 50.0, 62.5, 75.0, 100.0, 125.0, 150.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700.0, 800.0, 900.0, 1000.0, 1100.0, 1200.0, 1300.0, 1400.0, 1500.0, 1750.0, 2000.0, 2500.0, 3000.0, 4000.0, 5000.0
V Component of Current (CurV)	m/s	as above
Salinity (Salinity)	kg/kg	as above
Sound Speed (SndSpd)	m/s	as above
Water Temperature (WtrTemp)	K	as above

5.0 Federation Agreements

The JLVC Federation agreements document design rules, workarounds, and other necessities that are not enforceable in software. The agreements compliment the core standards and specification capability to ensure that each design is in compliance with the overarching standards.

Federation Agreements enable interoperability among JLVC simulation components participating in a federation execution. The agreements in this document are applicable across all exercises. Any agreements that are exercise-specific will be included in the event-specific information provided to participants.

5.1 JLVC Interface Standards

5.1.1 The Federation Object Model

To understand and interface with the JLVC Federation, it is imperative to know and understand the JLVC FOM. The JLVC FOM is an agreement amongst the JLVC federates on the data and datatypes to share on the HLA. The FOM builds on existing industry standards from SISO, JNTC, and Defense Information Systems Agency (DISA).

The JLVC FOM is derived from the Simulation Interoperability Standards Organization (SISO) Real-time Platform Reference FOM, Version 2 Draft 17.

JLVC uses the DoD HLA 1.3 set of HLA specifications. In particular, the following interpretations are held by JLVC regarding the information conveyed in the HLA 1.3 OMT. The letters “P,” “S,” and “N” in the PSCapabilities field in the HLA 1.3 OMT format FOM conveys specific meaning regarding a federate’s permitted publication and subscription behavior. The federate may publish to the object classes identified in the class hierarchy tables with a “P” (for publish). The federate may subscribe to the object classes identified with an “S” (for subscribe). The letter “N” indicates that the class is an abstract parent class and is not to be published nor created as that class. The federate shall not subscribe nor publish object classes identified with an “N” in the JLVC FOM.

Changes to the FOM will be needed as JLVC capabilities evolve. A goal of JLVC Interoperability is to minimize the impact of changes on JLVC stakeholders. Certain federate software design practices coupled with constraints on the FOM evolution process will help work toward this goal.

All proposed additions must be submitted to the JLVC Federation Manager for approval and incorporation into the FOM. The Federation Manager will typically approve addition of optional attributes to existing object classes or addition of optional parameters to existing interactions. Addition of new object classes or interaction classes, as well as associated Complex Data Types or Enumerated Data Types, are usually permitted

provided they do not duplicate model interfaces already expressed in the FOM. All other types of modifications will cause impact to existing integrated federates.

The current version of the JLVC FOM is dated 20 March 2009. It is available by request from Brian Gregg, mailto:brian.gregg@att.jfcom.mil The object classes and interactions used in JLVC may be seen in the publication and subscription sets shown as 7.5.2APPENDIX E: and 7.5.2APPENDIX F: respectively

5.1.2 HLA and the Run-Time Infrastructure (RTI)

Additional information may be obtained from the HLA Technical Library, http://www.msco.mil.

5.1.2.1 Federation Agreements for RTI Usage

The primary mechanism for intra-federation communications will be a version of RTI-NG Pro, using RID parameters added in support of MC02/DCEE/JNTC. As of September 2008, the current version in use by JLVC is v4.0.4. However, it is expected that JLVC will migrate to new releases (including ports to newer platforms, such as Redhat Enterprise Linux v5) as requirements dictate.

5.1.2.2 RTI Services

RTI NG Pro® implements version 1.3 (Draft 10, 2 April 1998) of the *High-Level Architecture (HLA) Interface Specification*.

RTI software is currently comprised of the RTI Executive process (RtiExec), the Federation Executive process (FedExec), and the libRTI library. Each executable containing federates incorporates libRTI. Federates may exist as independent processes or be grouped into one or more processes. A federate may simultaneously participate in more than one federation.

The *HLA Interface Specification* identifies the services provided by libRTI-NG to each federate and the obligation each federate bears to the federation. Within libRTI-NG, the class RTIambassador bundles the services provided by the RTI. All requests made by a federate on the RTI take the form of an RTIambassador method call. The abstract class FederateAmbassador identifies the callback functions each federate is obliged to provide.

While both RTIambassador and FederateAmbassador classes are a part of libRTI-NG, it is very important to understand that FederateAmbassador is abstract. The federate must implement the functionality declared in FederateAmbassador. An instance of this federate-supplied class is required to join a federation.

The *HLA Interface Specification* partitions the exchanges that take place between federate and federation into six management areas of the FedExec life cycle: Federation Management, Declaration Management (DM), Object Management,

Ownership Management, Time Management, and Data Distribution Management (DDM). Of these, the following are used in JLVC:

5.1.2.2.1 Federation Management

Federation management includes such tasks as creating federations, joining federates to federations, observing federation-wide synchronization points, effecting federation-wide saves and restores, resigning federates from federations, and destroying federations.

5.1.2.2.2 Declaration Management

Declaration management includes publication, subscription, and supporting control functions. Federates that produce object-class attributes or interactions must declare exactly what they are able to publish (i.e., generate).

5.1.2.2.3 Object Management

Object management includes instance registration and instance updates on the object producer side and instance discovery and reflection on the object consumer side. Object management also includes methods associated with sending and receiving interactions, controlling instance updates based on consumer demand, and other miscellaneous support functions.

5.1.2.2.4 Data Distribution Management (DDM)

DDM provides a flexible and extensive mechanism for further isolating publication and subscription interests, effectively extending the sophistication of the RTI's routing capabilities.

5.1.2.3 RTI configuration

5.1.2.3.1 Perfect Filtering. Perfect filtering will be disabled via a RID file parameter. Perfect filtering causes a significant increase in network traffic and is very CPU intensive for federates that make extensive use of regions for geographic filtering. However, federates will find that they will get object updates and interactions from outside the regions they subscribed to and will have to deal with that in the application. This happens because the exact region matching filtering in your local RTI component is replaced with multicast filtering at the NIC. The multicast filters are fixed regions and if any part of the application's subscription region overlaps a multicast region then the application will be subscribed to the entire multicast region.

5.1.2.3.2 Multicast Regions. RID file parameters will be set so that multicast region spanning publications will generate exceptions and duplicate updates/interactions will not be filtered by your local RTI component. Under normal RTI operation, publications without regions or with large regions will be duplicated for each multicast region in the space. For federations such as JLVC using large numbers of multicast addresses,

duplicated messages would be disastrous. If the application publishes something without a region or to a region that overlaps multiple multicast addresses, your local RTI component with throw an exception. To avoid this, everything must be published to a point region, i.e. lower bound = upper bound. With the sending of duplicate messages eliminated, the need for receiver filtering of duplicates is eliminated.

5.1.2.3.3 DDM Dimension Mapping. RTI-NG Pro can provide a listing of the mapping between DDM dimensions and multicast addresses. This provides an invaluable tool for debugging and data routing. This feature is controlled via a RID parameter (PrintMulticastChannelMapping), and is supported for both the Static Grid Partition DDM strategy, as well as the MC02 DDM Strategy. If the RID parameter is turned on, it will print out the mapping upon startup.

5.1.2.3.4 Connectionless State. The RID parameter RTIMode will be set to connectionless to allow the RTI to operate in a connectionless state. This prevents a crash or disconnection of one federate from affecting the rest of the federation. Coincidently, all network transport parameters must be set to UDPmulticast, (TransportMechanism UDPmulticast), as reliable Transmission Control Protocol/Internet Protocol (TCP/IP) communications are not supported in the connectionless RTI mode. Additionally there is no RTIEXEC or FEDEX process and no need for an RTI operator when connectionless mode is used. Federates are able to join and leave the federation at will. On the negative side, it means that there is no support for reliable transport, no RTI save, no sync points, and no advisories. RTI name uniqueness is not guaranteed by the RTI. Name uniqueness is managed by federation agreement. To work around the lack of advisories lazy discovery and timeouts will be used (advisories have been shown to be nonscalable).

5.1.2.3.5 Kernel Parameter Adjustments. The RTI will attempt to utilize more multicast addresses than most machines are typically configured to allow. Therefore, it is necessary to adjust the default max values for a few kernel parameters. For Redhat Linux versions 6.2 and greater, the sysctl command enables increase of default values (do man sysctl for more info). The safest and easiest approach is to add the following lines to the /etc/sysctl.conf file, which will configure the values correctly every time you boot. This increases the MAX values of various kernel parameters, NOT the default, so this will not have an effect unless the application asks the kernel to increase them:

```
# allow read and write buffers for sockets to be reasonable large

net.core.rmem_max = 2097152

net.core.wmem_max = 2097152

# increase number of multicast memberships per socket.

net.ipv4.igmp_max_memberships = 4096

# increase memory used for extra stuff per socket

net.core.optmem_max = 65536
```

Note: The above configures the default MAX size of send and receive buffers that an application can increase for a socket. It is important to increase these or the changes to dropping packets increases due to buffer overflow. For other operating systems, you will have to consult the man pages and/or documentation to see if a similar capability exists.

5.1.2.4 Available RTI Services for JLVC

The federation shall only call the following services to initiate federation events:

- Federation Management
  - Join Federation Execution
  - Resign Federation Execution:
    when a federate resigns, it can choose whether to include a directive to delete all objects for which it has delete privileges. With a connectionless RTI and timeouts, we no longer require explicit deletion.
  - The use of synchronization points is prohibited.
  - The use of the RTI save and restore services is prohibited. Federates will perform independent saves and restores.
- Declaration Management (DM)
  - Publish Object Class
  - Unpublish Object Class
  - Publish Interaction Class
  - Unpublish Interaction Class
  - Subscribe Object Class Attributes will not be used. Instead, use Subscribe Object Class Attributes with Region.
  - Unsubscribe Object Class
  - Subscribe Interaction Class will not be used. Instead, Subscribe Interaction Class with Region must be used.
  - Unsubscribe Interaction Class
- Object Management
  - Register Object Instance should not be used, instead use Register Object Instance with Region
  - Delete Object Instance should be used for Munition objects. All other object type instances should exist for the duration of federation execution.
  - Local Delete Object Instance
  - Update Attribute Values

  - Send Interaction shall not be used; instead use Send Interaction with Region.
  - Request Attribute Value Update
    This service shall not be used. All objects are being heartbeated with all their required parameters, thus the application should only have to wait from one or two heartbeats for data to arrive. If it is taking longer than that, then either the federate is bogged down, the network is congested, or packets are dropping. Sending out requests for more data will not help any of these situations.
  - Provide Attribute Value Update
    This request can be ignored.

- Ownership Management
  - The use of ownership management services is prohibited.

- Time Management
  - The use of time management services is prohibited.

- Data Distribution Management (DDM) will be used by all federates. Refer to Section 5.1.2.6 Scalability and DDM for JLVC for details on how DDM will be used for JLVC
  - Create Region
  - Modify Region
  - Delete Region
  - Register Object Instance With Region
  - Subscribe Object Class Attributes with Region
  - Unsubscribe Object Class with Region
  - Subscribe Interaction Class with Region
  - Unsubscribe Interaction Class with Region
  - Send Interaction with Region

- Support Services
  - The use of advisories is prohibited.

- Management Object Model (MOM) Services
  - MOM services are currently unavailable.

5.1.2.5 Secondary Communications Mechanisms

Federates are allowed to use additional communications mechanisms for information not contained in the core objects and interactions defined in this agreement. These additional communications mechanisms can take the form of private objects and

interactions within the primary RTI federation, secondary federations, or other unicast, broadcast or multicast messages. All high bandwidth secondary communications including encoded audio must take place on independent domains (LANs or Virtual Local Area Networks (VLANs)). DIS federates will communicate on separate domains (LANs or VLANs) interfaced to the primary federation domain via DIS gateways. WAN communications from Joint Training, Analysis, and Simulations Center (JTASC) will carry HLA traffic that will be converted to DIS via gateways at remote sites.

5.1.2.6 Scalability and DDM for JLVC

5.1.2.6.1 Introduction

5.1.2.6.1.1 Aggregates. The expected requirements for the JLVC Federation include being able to model a large number of individual entities, their command and control hierarchies, their emissions, and their tactical interactions. This is a significant load both in terms of the processing power that will be needed to simulate JLVC and the network bandwidth required to communicate between the federates. A number of approaches to scalability exist, and as a key factor in the success of JLVC, it is important to examine these approaches.

5.1.2.6.1.2 Update Rates. A traditional approach is to use aggregates to decrease the total object count; the Real-time Platform Reference (RPR) FOM BaseEntity.AggregateEntity object class exemplifies this approach. The problem with this class is that it precludes many sensor models from maintaining an entity level picture of the battlefield. Secondly, while some sensor models utilize templates to assume entity locations based on unit locations, multiple models using this capability send inconsistent entity locations to C4I systems. As a result, federations have used both aggregate and entity level representations. This combined approach increased object counts by one-third defeating the original purpose of decreasing the total object count.

Another approach is to limit the update rates on entity state data to the minimum level acceptable for model operation. We have relaxed our heartbeat rates from five seconds in standard DIS to sixty seconds. We have relaxed our dead reckoning thresholds from one meter to ten in position and from three degrees to fifteen in orientation.

5.1.2.6.1.3 Query Protocols. The third approach is to use query protocols. Query protocols require the observer to publish a representation of where he is looking so that the targets can respond with their information. They are useful for narrow beam searches and distributing sensor computation. Their primary drawback is latency. We decided to avoid this approach for the time being because it would require everyone to integrate new distributed sensor models into their federates.

5.1.2.6.1.4 Data Reduction. Standards such as the RPR FOM, on which the JLVC FOM is based, call for a large variety of data in case some application might require it. However, specific applications can trim away much of this data if their requirements can be met without it. For JLVC we have started the trimming by deleting radio receiver

objects and articulations from the FOM. Since nearly every vehicle has a radio receiver, that will reduce the object count. Articulations such as tank turrets are constantly moving and can generate unnecessary network traffic. Also, we made the collision interaction optional. However, the more we modify the FOM from RPR, the more development will be required by the federates and the harder it will be to make them participate in other federations. Since all the simulations are used in multiple programs, that is a significant concern. We will monitor traffic levels during our integration testing and evaluate objects and interactions that are causing high levels of traffic as candidates for change.

5.1.2.6.1.5 Partial Updates. The HLA allows use of partial updates and queries to avoid sending the same information repetitively. We make use of this capability to only send out position, velocity, and orientation information on dead reckoning threshold exceptions. For attributes that publish on parameter changes, we only send out the changed parameters.

5.1.2.6.1.6 Data Pre-distribution. As discussed in Section 4.4, the Joint Training Data Services (JTDS) will be used to generate scenario data that shows little, if any, change during scenario execution. This approach saves bandwidth and processing power.

5.1.2.6.1.7 Partitioning Introduced. The last approach to scaling we will discuss is partitioning. It relies on the assumption that not every federate needs to know the state of the entire virtual world. For example, if one federate is simulating entities in the southeast corner of the database, it may not need to know about what is happening in the northern and western quadrants of the database. Similarly, ground units may not need to know about emissions and high altitude bombers may not need to know about ground vehicles. If such partitions can be made, then enormous savings can be achieved by filtering out those objects before they reach the federate.

This section focuses on partitioning as implemented in JLVC. This information is introductory in nature and DDM experienced and/or impatient readers can skip to the last two sections focusing on the DDM approach to JLVC.

5.1.2.6.2 Partitioning

The concept behind partitioning is to assign the simulation of objects to federates in such a way as to minimize their requirements for information from other federates. This requires mechanisms for delivering only relevant knowledge to each federate. Before examining these mechanisms, let us examine where the obstacles to scalability in a federation lie. Large number of remote objects and interactions can cause federates to fail by:

- Exceeding internal object storage, or making object lookup too slow. When the number of objects a federate keeps track of internally gets too large, a number of problems can happen. Fixed arrays can overflow. Process size can get too large and cause swapping and poor performance. More subtle are computations that grow faster than linearly with the number of entities. As the number of

entities increases beyond a few hundred or thousand, these processes can slow down the federate to the point that it cannot maintain real-time.

- The time required to process network state updates can grow to the point that there is insufficient time to process internal state updates again causing the simulation to slow down to slower than real-time.
- Too much time between the processing external state updates and interactions can cause kernel buffer overflows. As data comes in from the network, it is stored in buffers in the operating system. If the federate can't empty the kernel buffers fast enough, some network data will be lost because the kernel has no place to put it.
- If the number of packets per second arriving from the network exceeds the capacity of the network interface card to process them, some data can be lost.
- If the network infrastructure components cannot handle the traffic levels, data can be lost. These components include hubs, switches, encryption devices, WAN interfaces, and WAN pipelines. In recent years, the progress that has been made in networking has made the network components very tolerant of high bandwidth traffic provided you can install adequate connections to your sites and pay for the bandwidth. We may very well find that we do have WAN bandwidth problems for JNTC, but they will not be due to limitations in technology.

These problems can be avoided if the network and scenario structure can be arranged so that each federate only needs to listen part of the total traffic resulting in a scalable simulation. Traffic can be partitioned in many different ways. Geographic filtering limits information from entities located far away in the simulation playbox. Domain filtering limits information from vehicles in other domains. Radio traffic can be filtered by dividing up radio signal interactions by frequency. Attribute filtering can eliminate attributes that a federate isn't interested in. By looking at each simulation's models and determining what information the simulation needs to operate effective partitioning schemes can be developed.

Once an effective partitioning scheme has been developed, appropriate mechanisms are required to take advantage of it. There are a number of places where traffic can be filtered and they roughly correspond to the different layers where scaling problems can be found.

- Filtering can occur at the application level. As the federate gets new data from remote federates via the RTI, it can throw away those that it finds uninteresting. This helps with maintaining reasonable sizes for the federate's internal tables.

- Filtering can be done at the RTI. This is essentially the same as filtering at the application level except that it is done by RTI code rather than by specialty code in the federate application. This has a benefit for the RTI internal tables in addition to the benefit for the applications tables. It also cuts down on the callbacks required to transfer data from the RTI to the application code.

- Filtering can be done at the Operating System level. This can provide a benefit to the kernel buffers as well as to the RTI and application. It also may result in fewer process interrupts to tell the application to process new network data.

- Filtering can be done at the Network Interface Card (NIC) Level. This can shield that application from operating system interrupts, prevent kernel buffer overflow, and provides all the benefits of the previous approaches.

- Modern network switching hardware provides mechanisms to isolate computers so that each is effectively on its on network. The send and receive lines are separated so that sends don't collide with receives. Thus, a computer could potentially be able to take advantage of the entire 100 Mbits/second commonly provided by Ethernet. Buffers in the switch ports, with synchronized feeds to a much higher speed backbone, prevent collisions and saturation on the backbone. Modern switches also have a filtering capability on their output ports. This capability allows us filter traffic not desired for a particular host before it gets to the NIC. This can be useful if the switch has more filtering capability than the NIC.

- The hardware that links the LAN with Wide Area Network (WAN) circuits can be set to filter information that doesn't need to go from one site to another.

- The last place that one can filter output is on the side of the generating federate. If that federate knows that no one is interested in the information it is publishing, then it can turn off the publication of that information. This can be done at either the application or the RTI layer.

Filtering at the NIC is the first level at which filtering is free to a federate. That is because the filtering is done in hardware and the CPU of the federate is not interrupted to do the processing. Thus, anything filtered at the NIC has no impact on the processing capacity of the federate. Also, to set up filtering requires some communications back through the network. Communication with the NIC is very cheap and reliable. The relative bandwidth capacity and filtering capability of the NIC and the switch determines whether switch filtering is better than NIC filtering. If the total traffic on the network backplane exceeds the bandwidth capability of the port to the federate or the NIC, then filtering is best done at the switch. In the STOW 97 exercise, switch filtering was used because the federates only had 10Mbit/sec ports and NICs, in AO00 NIC filtering was used because the NIC allowed subscriptions to change faster than the switch. The presence of WAN bottlenecks may require special filtering solutions at the WAN interfaces. Source filtering has the benefit of totally removing any information not required by other federates from the network. It is the most effective solution if other federates do not require the information. However, if even one federate requires the information then source filtering is unsatisfactory. Point-to-point connections may be used to mitigate this problem, but the number of point-to-point connections increases very rapidly with the number of federates so such schemes are only feasible for small federations.

5.1.2.6.3 Multicast

In addition to its' own processing, there are two primary filtering tools available to a federate, the first is the RTI itself and the second is Internet Protocol (IP) addressing. The RTI requires every federate to notify it of its interests via subscriptions. This allows the RTI to support filtering either on the transmitting RTI or at the receiving RTI. IP addressing allows filtering to be done by the network hardware, which is generally more efficient. IP addressing takes several forms. In unicast addressing, information is sent specifically to a single target machine using its IP address. From a filtering point of view, this is the best form of filtering since you know exactly what host the information is going to and can send it only the information it needs. It is the underlying transport mechanism when reliable transport is specified for an attribute. However, unicast addressing is inherently unscalable. This is because the majority of attribute updates and interactions are subscribed to by many federates. Unicast addressing requires that a separate copy of the data be sent to every machine that is interested in it. Thus the bandwidth to transmit a piece of information to n machines via unicast is n times as great as the bandwidth to transmit it to one machine, plus the sending machine has to send the message n times. To avoid this problem, DIS replaced the use of reliable protocols with best effort transmission using broadcast addressing. Everyone on the local domain gets every data transmission. This is also inherently unscalable. To overcome this DIS simulations are often setup to use multiple domains, and we will use that strategy with our DIS federates. However, carrying this beyond a few domains is hard since each domain requires a gateway and the scheme is static so that while it may be optimal for one part of the scenario it may become very inefficient as the positioning of the units changes. Fortunately, the IP provides us with a third addressing mode that is similar to creating thousands of independent broadcast subnets in a single domain. This is multicast addressing; it is the primary partitioning mechanism used for JLVC. IP provides a range of addresses that do not correspond to any particular machine. Instead, any machine that subscribes to an address in this range hears all the traffic send out on that address. Machines can turn information from a multicast address on and off dynamically by telling the operating system what addresses they are interested in. Multicast addresses are also called multicast groups. Once the operating system knows what multicast groups you are interested in it tells the NIC and the switch the same information. This allows filtering to occur at the NIC and/or at the switch. Older NIC cards had very limited filtering capabilities, if you needed to listen to more than 16 multicast groups, the card would essentially stop filtering and require the operating system (OS) kernel to do the filtering using the machine's CPU. Some cards can now handle thousands of subscriptions without relying on the OS. Similarly, modern switches can also handle thousands of multicast groups and can also pass the filtering information on to the WAN interface without any additional work on the part of the application or RTI.

5.1.2.6.4 Declaration Management

The RTI has two mechanisms for filtering or partitioning data: DM and DDM. DM is based on object and interaction classes. Class subscriptions allow federates to tell the

RTI which classes interest them. For example, one federate can subscribe only to the Aircraft class while another can subscribe only to the GroundVehicle class. This type of filtering in the RTI can be implemented by receiver filtering and source squelching. Because of the way that the HLA specification is written, it is difficult to effectively use multicast to support class-based filtering when class hierarchies are used. For example, if both GroundVehicle and Aircraft inherit their attributes from a common super class so there is no way to send their updates to different spaces. If no federates are interested in GroundVehicle information then the RTI can tell a federate(s) not to publish it and it would be very efficiently squelched at the source. However, if one federate, perhaps a logger, which needed to record GroundVehicle updates, actively subscribed to it, every federate would still get the update packets passing through their NICs, their OSs, and into their RTIs. Once the updates reached the RTI they would get thrown away, but the impact would have already been felt on performance. Another problem with the class-based approach to data partitioning is that it is static and does not support dynamic filtering where the information a federate requires is based on the current state of its objects. For these reasons, the JLVC Federation will rely on Data Distribution Management (DDM) for filtering.

5.1.2.6.5 Overview of DDM

This section describes Data Distribution Management as defined in the RTI-1.3 specification. While the IEEE 1516 specification has modified DDM, our approach is compatible with both. DDM uses two addressing concepts to determine how to route attribute updates and interactions from one federate to another. The first is the concept of discrete spaces. You can define spaces in the FED file and associate attributes and interactions with them. Unfortunately, the RTI-1.3 spec requires that each attribute class or interaction class is associated with one and only one space and that association has to be specified in the FED file. Thus the association between attributes/interactions and spaces is static. This severely limits the filtering capability of spaces. Also, spaces are eliminated in 1516. As a result, the JLVC DDM approach uses only one space called HyperSpace.

The second addressing concept is based on a continuous spatial analogy where each space is associated with a set of dimensions. Each space has to have one or more dimensions. Examples of possible spaces and dimensions are: "3DCartesianSpace" with dimensions, x, y, and z; "RadioFrequencySpace" with dimension frequency. Since this is a continuous addressing concept, an address has to be specified in terms of subspaces defined by upper and lower bounds on each dimension. A pair of upper and lower bounds is called a range. An extent is a hyper-rectangle defined by one range on each of the dimensions of the space. A region is a set of extents defined by the *Create Region* service. Regions are the basis of DDM addressing and they are used for both publishing and subscribing to attribute updates and interactions.

Regions may be dynamically modified using the *Modify Region* service. The services *Register Object Instance With Region* and *Associate Region For Updates* allow the federate to direct the RTI to send attribute updates through a region. The service *Send*

Interaction With Region is used to direct the RTI to send interactions through a region. To make this useful, corresponding services allow federates to subscribe to attribute updates and interactions within these regions. An update region is defined to overlap a subscription region if and only if the regions use the same routing space and the corresponding extent sets overlap. Overlap requires that all dimensions have at least one subscription range and update range where the lower bounds of each range are less than the upper bounds of the other range or where the lower bounds are equal. If the regions overlap the data published to update region will be received by any federate subscribing to the subscription regions.

5.1.2.6.6 Linking DDM to Multicast

The link between DDM regions and multicast addresses is not defined by the RTI specification. It is an RTI implementation specific interface. RTI-NG Pro allows you to specify how many multicast addresses you want to allocate to each dimension of a space and the start of the multicast range you are using. Multicast addresses are assigned to extents automatically by internal RTI algorithms. The total number of multicast addresses allocated to a space is the product of the number of multicast addresses on each dimension. Each dimension is divided uniformly into an extent for each multicast address. Thus, the entire space is uniformly divided into fixed extents, one per multicast address.

5.1.2.6.7 Application Spaces

The DDM approach we use for JLVC is based on the concept of application spaces. The approach is to use one dimension to create subspaces within an RTI space. Attributes can then be published to subspaces that change over time, within the single RTI space. This approach is named the application space approach, because the spaces are defined by the needs of the application. It is unfortunate that this concept overloads the name "space" when the RTI is already using it. The following description of application spaces is from [APPENDIX B: References, B.3 Technical, 8]:

An application space is defined as a subspace of an RTI routing space and an associated list of object classes. Each dimension in an application space is mapped to a dimension in an RTI routing space. All dimensions within an application space are mapped to the same RTI routing space. Every object class associated with an application space must have all of its attributes bound to the RTI routing space to which the application space is mapped. This approach allows the federate to treat spaces in any way that is convenient for internal use, and then map the application spaces to a more complex RTI routing space that is hidden from the application.

As an example, consider a simulation that supports fixed wing aircraft, tanks with a speed sensitive sensor model (perhaps a Doppler radar) and radios. So, for the federate, it is convenient to have the application spaces and dimensions defined as listed in Table 8.

Table 8. Example Application Spaces and Associated Dimensions

Air Space	Ground Space	Radio Space
Latitude	Latitude	Frequency
Longitude	Longitude	
Altitude	Speed	

These application spaces can be mapped into a single RTI routing space by creating a space with the dimensions described in Table 9. So, the RTI routing space would have five dimensions defined as listed in Table 9. In the above example, the application maps the air space related dimensions (latitude, longitude and altitude) to the appropriate RTI Routing Space dimensions. For subscription regions within air space, the extents associated with dimensions not in air space are set to the minimum and maximum extent. This prevents filtering from being performed on irrelevant dimensions. The same is done for ground space and radio space.

Table 9. Associated RTI Routing Space and Associated Dimensions

RTI Routing Space
Latitude
Longitude
Altitude
Speed
Frequency

The critical factor that makes this effective is a feature provided by DDM. The interface specification allows the federate to create a subscription region and instruct the RTI to filter out all but a given list of object classes and interaction classes for that region. When a subscription region within an application space is created, a region in the RTI routing space is really created, but the RTI is instructed to inform the federate only of the object classes and interaction classes that are associated with that application space. With this scheme, it becomes possible to have an air space and a ground space and to allow different object classes to be associated with these two spaces even though they share a common inherited attribute

One of the goals of this effort was to reduce the number of subscription regions. Application spaces that share similarly sized regions can effectively be merged into a single region. This is possible because the application spaces are all mapped to a single RTI routing space and hence a single region can exist in multiple application spaces. Therefore, the application space DDM scheme provides the ability for the application to optimize the use of regions and still maintain the flexibility to create the

desired application spaces. Using this approach, the number of regions can be reduced significantly.

5.1.2.6.8 RTI Services

This section expands on the similar section 5.1.2.4 Available RTI Services for JLVC. The following Data Distribution Management related functions should be used in place of their counterparts (ie. those without "WithRegion"):

- createRegion
- notifyAboutRegionModification
- deleteRegion
- registerObjectInstanceWithRegion
- associateRegionForUpdates - this service allows you to associate a region to an already registered object. This is much less efficient than using the corresponding registerObjectInstanceWithRegion call. So, if possible, please register your objects with a region.
- subscribeObjectClassAttributesWithRegion
- unsubscribeObjectClassWithRegion
- subscribeInteractionClassWithRegion
- unsubscribeInteractionClassWithRegion
- sendInteractionWithRegion
- requestClassAttributeValueUpdateWithRegion

Using registerObjectInstanceWithRegion to associate a region to an object upon its registration is more efficient than the two-step process of using registerObjectInstance and associateRegionForUpdates. Therefore, please try to register your objects with a region.

ALL regions used for publishing data **MUST** be point regions. In essence, this means that the upper and lower bound in each dimension are equivalent.

For subscription regions, the lower and upper bound can be anything you want them to be (as long as they are valid). You can create a subscription region that covers the entire range from MIN_EXTENT to MAX_EXTENT in each dimension if you so desire.

You may reuse regions for more than one object, interaction, etc. The only catch is that if you reuse a region for both publication and subscription, it has to be a point region.

You may create multiple extents in each region if you wish. We have found it easier to simply create multiple regions instead since its easier to manage (and the number of extents per region can't be changed once the region is created).

When maintaining a region for an object that is "moving" (and therefore, the region has to "move" with it), a thresholding scheme will be used. Basically, there are two parts to make this work:

- When publishing, you update the region periodically as the object it is associated with moves. However, to limit the number of times the RTI is notified, you should only call notifyAboutRegionModification when the object has moved more than the given threshold for that appspace as described below.

- When subscribing, you should increase the ranges in the latitude and longitude dimensions by the same threshold used by the publishers.

You must associate every attribute you plan to send to a region. One common mistake is to only associate the attributes that you actively update to a region. If you have attributes that you would send if someone asked for them (ie. optional attributes that you default, or don't really model, etc), you have to make sure that those attributes are associated to the region as well.

Every federate in JLVC must support DDM and use the appropriate "WithRegion" services in the RTI. Due to many factors, we can't mix and match DDM with DM and still maintain performance and scalability. For example:

- If you publish an attribute not associated to a region, the RTI will send it to EVERY multicast group that it is using. This could result in thousands of messages being sent on the wire for every single update or interaction sent.

- If you subscribe without a region, the RTI will subscribe to every multicast address its using (again, it could be thousands). This will cause you to receive EVERY message that is sent regardless of whether or not you want it. Further, without perfect filtering turned off, the RTI won't filter anything and your application will receive it. For JLVC, you MUST subscribe with a region. This is because the RTI assigns multicast addresses uniformly to each app-space. Let's say we have 100 multicast address assigned to geographically filter the red ground vehicle space. Then the damage assessment space also has 100 multicast addresses assigned to it. However, we only want to use one of those spaces. If everyone who subscribes to the damage assessment spaces uses a region with the second and third dimensions set to (MIN_EXTENT, MIN_EXTENT) then only one of the 100 addresses will be used. The remainder will be thrown away and not count against our multicast address budget.

5.1.2.6.9 JLVC DDM Scheme

For JLVC, we will use one RTI space called HyperSpace. On top of HyperSpace we have added a layer of mapping to produce "application spaces" which are implemented as an enumerated mapping to a dimension within an RTI space. The second piece of the application space idea is that the other dimensions of the RTI space change their

meanings based on which application space is in use. For instance, some application spaces are geographically mapped, with two other dimensions being latitude and longitude, and other application spaces are enumerated values, with only one dimension in use with min and max bounds specified specifically for that space.

In the case that a dimension is not used, we set its lower and upper bounds to MIN_EXTENT.

These application spaces are contained in the RTI space called HyperSpace (the only one supported) which has three dimensions, "appspace", "one", and "two". The appspace dimension is enumerated and split into N bins (as defined by the NUM_APPSPACES macro in the following code). The semantics of the "one" and "two" dimensions is dependent on the specific appspace with which they are assosicated.

It is important to utilize the correct number of appspaces (defined by the macro NUM_APPSPACE), the extents of the geographic regions (defined by the macros LATITUDE_LOWER/UPPER_DIMENSIONS and LONGITUDE_LOWER/UPPER_DIMENSIONS), and the actual numbers assigned to each appspace. We will try to keep the appspace names (and their numbers) as consistent as we can to minimize the effects on each federate developer.

First, here are some helper functions that convert from appspace values to RTI extent values, and are used in the example code snippet below.

```
//*****************************************************************
//
// first, a couple of helper functions to convert "internal" values
// to RTI extent values...  these three functions correspond to the
// possible algorithms "enumerated", "linear" and "partitioned".
//
//*****************************************************************

// a function for translating an enumerated value into an RTI extent value
// that is in the range MIN_EXTENT and MAX_EXTENT.  These MIN/MAX extents are
// defined in the RTI header files.
//
// this function can be used for any appspace that is "enumerated".
RTI::ULong map_enumerated(
    int ev,         // a value such that min <= ev <= max
    int min,
    int max)
{
    // sanity check.
    if ((ev > max) || (ev < min))
      die_a_horrible_death();

    // scale indicates how "big" each bin is when we divide up the RTI
    // extent range..  The addition of one to the denominator adjusts
    // for the fact that we are mapping into a closed range (min <= ev
    // <= max) and therefore the number of bins in the range is one
    // more than the difference..
```

```
    double scale = (double)(MAX_EXTENT - MIN_EXTENT) /
                   (double)(max - min + 1);

    // now, the value we are looking for is the value exactly in the
    // center of the appropriate bin.  the last part of the following
    // equation (scale/2) moves the value into the center of that bin.
    double value = MIN_EXTENT + scale*(ev-min) + scale/2;

    return((RTI::ULong)value);
}


// a function for translating a value from an interval value to
// an RTI extent value that is in the range MIN_EXTENT and
// MAX_EXTENT.  These MIN/MAX extents are defined in the RTI header files.
//
// this function can be used for any "linear" appspace.
RTI::ULong map_linear(
    double ev,       // a value such that min <= ev <= max
    double min,
    double max)
{

    // Crop to valid range..    for linear mappings, we clip all
    // values outside the range to either the min or max..
    ev = (ev > max) ? max : (ev < min) ? min : ev;

    // scale in this case represents a scaler used to put the value
    // in the right place in the linear mapping..
    double scale = (double)(MAX_EXTENT - MIN_EXTENT) /
                   (double)(max - min);

    double value = MIN_EXTENT + scale*(ev-min);

    return((RTI::ULong)value);
}
```

As an example, let's examine publishing a friendly aircraft which is at the position (22.43, -110.22) in degrees lat/long:

The interesting bits for this case are:

```
// The next few lines define some "macros." The macros are only used for
// convenience in this example and do not necessarily represent good coding
// practice.


// This macro defines the number of appspaces that exist in the current DDM
// strategy.  This value is federation specific and may change over time.
// It is suggested that you have the flexibility to modify this value without
// having to modify code.  This value is derived from the RTI's RID file
// parameter NumberOfSubspacePartions.  This value used is 0-based and
// therefore the range is from 0 to NUM_APPSPACES-1.
```

```
#define NUM_APPSPACES 200


// these values for the latitude and longitude need to cover the whole
// playbox..  In other words, a federate does NOT adjust these even if
// their terrain database is only a subset of this area.
//
// this values below are in degrees, using the following logic:
//
//   the origin is the intersection of the equator and the prime
//   meridian and is represented as (lat=0.0, long=0.0)
//
//   a latitiude value ranges from -90 (south pole) to 90 (north pole)
//   a longitude value ranges from -180 (west of PM) to 180 (east of PM)
//
// so, a pair like (lat=30, long=50) means 30 degrees north of the equator
// and  50 degrees east of the prime meridian.
//
// another example, (lat=-30, long=-125) means 30 degrees south of the
// equator, and 125 degrees west of the prime meridian.
//

#define LATITUDE_LOWER_DIMENSION 30.0
#define LATITUDE_UPPER_DIMENSION 42.0
#define LONGITUDE_LOWER_DIMENSION -125.0
#define LONGITUDE_UPPER_DIMENSION -109.0


// These values are found in the RTI's RID file and are federation-specific.
// In general, these values should be parsed out of the RID file.  Here,
// we are using macros for convenience.
#define BLUE_AIR_SPACE 1
#define RED_GROUND_SPACE 8
```

So, to publish a blue aircraft into the "blue_air_space", with a position of (22.43, -110.22) in degrees lat/long, you would create a region using the following values:

```
    RTI::ULong app = map_enumerated(BLUE_AIR_SPACE, 0, NUM_APSPACES-1);
    RTI::ULong lat = map_linear(22.43,
                                LATITUDE_LOWER_DIMENSION,
                                LATITUDE_UPPER_DIMENSION);
    RTI::ULong lng = map_linear(-110.22,
                                LONGITUDE_LOWER_DIMENSION,
                                LONGITUDE_UPPER_DIMENSION);


    RTI::SpaceHandle sh;
    RTI::DimensionHandle dh;
    RTI::Region *mRegion;
    RTI::ULong bound;

    // create a region.  assume its "part" of the object it is for.
    try {
```

```
        // get the handle for the RTI routing space.  For JLVC,
        // this will always be for "HyperSpace".
        sh = raf->getRoutingSpaceHandle("HyperSpace");

        // create a region..  assume the simple case of just one extent
        // for this region.  you can have more than one extent per
        // region if you want.
        mRegion = raf->createRegion(sh, 1);

        // first dimension is the appspace.
        dh = raf->getDimensionHandle("subspace", sh);
        mRegion->setRangeLowerBound(0, dh, app);
        mRegion->setRangeUpperBound(0, dh, app);

        // second dim is the lat
        dh = raf->getDimensionHandle("one", sh);
        mRegion->setRangeLowerBound(0, dh, lat);
        mRegion->setRangeUpperBound(0, dh, lat);

        // third dim is the lng.
        dh = raf->getDimensionHandle("one", sh);
        mRegion->setRangeLowerBound(0, dh, lnt);
        mRegion->setRangeUpperBound(0, dh, lng);

        // let the RTI know the region changed.
        raf->notifyAboutRegionModification(*mRegion);

        // use mRegion with registerObjectInstanceWithRegion so
        // that the object is associated to a region upon creation.
        // This is better on the RTI and federation than registering
        // the object and then associating it with
        // associateRegionForUpdates

        // now, you can call updateAttributeValues as normal...
    }
    catch(RTI::Exception &e) {
        cerr << "exception: " << &e << endl;
    }

// now, as your object moves, you also have to update the region
// (ie. setRangeLowerBound, setRangeUpperBound, and then
// notifyAboutRegionModification) *before* you call updateAttributeValues()
```

A similar code snippet that walks through the creation of a subscription region is below. It utilizes the same helper functions from the above code snippet:

```
// a simple example to set up a subscription region for blue aircraft
// in the given lat/lng box..


    RTI::SpaceHandle sh;
    RTI::DimensionHandle dh;
    RTI::Region *mRegion;
    RTI::ULong bound;
```

```
try {

    // get the handle for the RTI routing space.  for JLVC,
    // this will always be for "HyperSpace".
    sh = raf->getRoutingSpaceHandle("HyperSpace");

    // create a region..  assume the simple case of just one extent
    // for this region.  you can have more than one extent per
    // region if you want.
    mRegion = raf->createRegion(sh, 1);

    // the min and max values used below in the various
    // calls to map_enum and map_linear should probably be defined
    // in a non-hardcoded way.  The extents of the terrain may
    // change over time, so you should have the ability to adjust
    // these easily..  similarly, the appspace range
    // may also change as more appspaces are added..   so, plan
    // on these values being in flux.  They are show below hardcoded
    // just to make the example easier to understand.


    // blue_air_space is 1, as defined in RID file.  if we wanted
    // to create a region for red_air_space, this value would be
    // 2.  similarly, for aggregate_space, this value would be 10.
    //
    // the values 0 and 30 come from the APPSPACE_DIMENSIONS
    // macro.
    bound = map_enum(BLUE_AIR_SPACE, 0, NUM_APSPACES-1);
    dh = raf->getDimensionHandle("subspace", sh);
    mRegion->setRangeLowerBound(0, dh, bound);
    mRegion->setRangeUpperBound(0, dh, bound);


    // subcribe to the latitude range 33.5555 to 38.0
    // the values for min and max for "latitude" will come
    // from the LATITUDE_ LOWER/UPPER_DIMENSIONS macro
    dh = raf->getDimensionHandle("one", sh);
    bound = map_linear(33.5555, 30.0, 42.0);
                       LATITUDE_LOWER_DIMENSION,
                       LATITUDE_UPPER_DIMENSION);
    mRegion->setRangeLowerBound(0, dh, bound);
```

```
    bound = map_linear(38.0, 30.0, 42.0);
                       LATITUDE_LOWER_DIMENSION,
                       LATITUDE_UPPER_DIMENSION);
    mRegion->setRangeUpperBound(0, dh, bound);


    // subscribe to the longitude range from -113.1234 to -110.0
    // the values for min and max for "longitude" will come from
    // the LONGITUDE_LOWER/UPPER_DIMENSION macros
    dh = raf->getDimensionHandle("two", sh);
    bound = map_linear(-113.1234,
                       LONGITUDE_LOWER_DIMENSION,
                       LONGITUDE_UPPER_DIMENSION);
    mRegion->setRangeLowerBound(0, dh, bound);
    bound = map_linear(-110.0,
                       LONGITUDE_LOWER_DIMENSION,
                       LONGITUDE_UPPER_DIMENSION);
    mRegion->setRangeUpperBound(0, dh, bound);


    // tell the rti we have modified the region..
    raf->notifyAboutRegionModification(*mRegion);


    // .. assume that objClassHandle and attrHandleSet are set up..

    raf->subscribeObjectClassAttributesWithRegion(
        objClassHandle,
        *mRegion,
        *attrHandleSet,
        RTI::RTI_TRUE);

}
catch(RTI::Exception &e) {
    cerr << "exception: " << &e << endl;
}
```

5.1.3 Coordinate Systems

5.1.3.1 Geocentric Coordinate System

All network world position representations shall use the Geocentric Coordinate System (GCC), defined as a coordinate system with origin at the center of the earth, Z axis projected through the North Pole, X axis projecting through zero degrees latitude and zero degrees longitude, and the Y axis projecting through 90 degrees East longitude and 0 degrees latitude. Distances are measured in meters.

5.1.3.2 Entity Coordinates

Standard DIS defines the position of an entity as the position of its center of mass. While this is natural for aircraft, it causes extra computation for ground and surface

entities. To save that computation the meaning of a vehicle's position and orientation shall be:

- For aircraft and subservice vessels, the z origin of their coordinate systems shall be at the center of mass of the aircraft.
- For ground entities, the z origin of their coordinate systems shall be located at the bottom of the object.
- For surface vessels, the z origin of their coordinate systems shall be at the nominal waterline of the entity.

Angular orientation shall be represented by three angles (Euler Angles) defined as follows. All rotations should be done in order presented here.

- yaw - rotate around vertical axis through the vehicle. Positive rotation is clockwise when looking down at the vehicle.
- pitch - rotate around axis through the width of the vehicle. Positive rotation is clockwise when looking out the right side of the vehicle.
- roll - rotate around axis through the length of the vehicle. Positive rotation is clockwise looking out the front of the vehicle.

5.1.4 Kinematic Attributes

The following attributes will be used to represent an object's kinematic state. These attributes will always be updated atomically even if only one requires an update.

- Position: x, y, z (meters)
- Velocity: x, y, z (meters per second)
- Orientation: yaw, pitch, roll (radians)

5.1.4.1 Dead Reckoning

Federates shall use dead reckoning (DR) to reduce the rate at which kinematic data must be updated between federates. Each federate shall maintain a dead reckoning position model for entities. This model shall be used to update the position of remote entities when performing calculations with the entity's position data. For its own objects, the federate will periodically compare the true position with the dead reckoned position and if the errors exceed the thresholds defined here, all kinematic attributes shall be updated.

Federates shall support the DRM (Static) and DRM (FPW) algorithms. If a federate wants to use a DR algorithm other than DRM (Static) or DRM (FPW), please inform the JLVC Federation Manager so that they can assess the impact (if any) on the federation.

5.1.4.2 Update Thresholds

Federates shall update an entity's Spatial attribute when the error between its dead reckoned position and true position exceeds:

(a) Position, the lesser of 10.0 meters or 100% of entity size along each dimension.

(b) Velocity, is unlimited

(c) Orientation, error around any axis exceeds 15 degrees

Only entities will be updated on DR threshold exceptions. All other objects such as aggregates will only be updated on heartbeats and change events.

5.1.4.3 Ground Clamping

No ground clamping is required of any federate.

5.1.5 Data Encoding

5.1.5.1 Simple Data Types

For simple data types, use those provided by the OMT with the exception of "any" which should not be used. The size of the data types in 8-bit bytes are shown in Table 10.

Table 10. Simple Data Type Sizes in 8-bit Bytes

Data Type	Number of Bytes
boolean	1
char	1
octet	1
short	2
unsigned short	2
float	4
long	4
unsigned long	4
double	8
long long	8
unsigned long long	8
string	string length + 1, NULL terminated

5.1.5.2 Byte-Order

Federates shall use Big Endian byte ordering for attribute and parameter data types on the network.

5.1.5.3 Data Alignment

All data types will be aligned on natural boundaries, with explicit padding filling in any gaps. Thus, byte long data elements should lie on byte boundaries, and 8-byte data elements should start on addresses that are a multiple of 8 bytes. Strings are considered an array of 1-byte elements; therefore, they can occur at any byte address. For more detail, refer to the GRIM for the RPR FOM [APPENDIX B: References, B.3 Technical, 9].

5.1.6 Entity Type Mappings

All legal entities and munitions are listed in the JLVC training enumerations reference. If your sim needs to simulate entities or munitions not on the list, or you feel that there are problems in the current list, please contact Gary Redinius at mailto:gary.redenius@ATT.JFCOM.MIL with your recommendations.

5.1.7 Object IDs

Each federation object must have a unique RTI identifier. This identifier must be specified by the application when registering the object. The application is responsible for making sure that the ID is federation unique. If possible federates are encouraged to maintain the same name for each object across save and restore. To do this and to allow monitoring of entities by source simulation, most RTI object names will start with the first 2 characters of the name of the simulation producing the object followed by the IP address of the machine simulating the object:

<application name>/<IP address in hex>/<your locally unique id here>

for example:

AW/c0a89d0e/562

JC/c0a69d0f/A10345

The RTI object identifier shall not exceed 48 bytes.

- <application name> must come from the following list of names. If your apppication does not have a name here please contact the Federation Manager to get it added. Except in cases of duplication, the first two letters of the name are used as the application name. That way monitor applications can determine which simulation the entity comes from by only doing a string compare on the first two characters. Case is significant. In fact if you are of a mind to be miserly with bandwidth, you can send out just the first two characters of the sim name; that is the minimum requirement.

  - **AW**SIM
  - **JC**ATS
  - **JS**AF
  - **JL**OD
  - **JD**LM
  - **HD**C
  - JLVC**DT**
  - **MT**WS
- <IP address> The IP address will be 8 characters expressed as hex digits without any dots. So the IP address 192.168.157.14 will be represented as c0a89d0e.
- <your locally unique ID here> Each simulation must generate its own locally unique character string.
- DIS entities have no RTI names so the only way to identify their source is to use the site and application part of the DIS ID. It is also necessary to use these fields to keep the DIS IDs unique. The IDs provided by the DIS simulators will be used for EntityIdentifier attribute of the BaseEntity class. And vice versa. These identifiers have the format site id, application id, entity id. Site IDs. Please use the site numbers below when publishing objects. Each simulation is responsible for ensuring that their application id is selected so that all their ids are unique. Simulations at Distributed Mission Operations Center (DMOC) will all use the same site id for technical reasons related to their simulations.
  - 1. TENCAP MUSE
  - 25. ITM (NTC Live units)
  - 26. (All virtual/constructive at 19th SOS)
  - 48. All virtual/constructive players at DMOC Kirtland Air Force Base (AFB)
  - 88. EP3
  - 96. JLOD
  - 100. AWSIM
  - 106. JCATS
  - 107. ACE-IOS
  - 109. NWARS-NG
  - 119. NETWARS
  - 123. MTWS
  - 201/202/203/204 JLOD
  - 525. TENA

– JSAF (JSAF site identifiers are the 3rd octet of the site's IP address and the app ID is the 4th octet of the particular JSAF machine that is generating the entity)

5.1.8 Unique Identifiers

Each federation object must have a unique JLVC identifier (JLVC ID). This identifier must be specified by the OBS application xml file, and consist of 11 characters. The OBS application is responsible for making sure that the ID is federation unique. Federates are required to maintain the same JLVC ID for each object across save and restore.

Each federate also needs to provide a unique identifier between a fire and detonate event in order to be able to capture this data for After Action Review (AAR) purposes.

Marking Text (Bumper Number). The JLVC will continue to link with DIS-based simulations. DIS simulations utilize a combination of DIS Entity ID and Marking Text to ensure unique identifiers for each entity in the federation. JLVC follows the RPR 2.0 GRIM for Marking Field so its length is 32 characters. When linking JLVC with DIS simulations the HLA-DIS translators will publish the JLVC ID in the DIS Marking Field on the DIS side. The DIS simulation's Marking Text will be published on the HLA side as the 11 character string in the HLA entity update.

5.2 Model Characteristics

5.2.1 Attrition

The JLVC Federation employs the convention that the shooter federate determines whether or not a fired munition hits the target (or for indirect fire detonates at a specified lat/long) while the target federate determines the damage it sustains if there is a hit. In terms of FOM constructs, the shooter federate initiates a WeaponFire and/or MunitionDetonation interaction(s). The WeaponsFire interaction will include an Event ID with matching Event ID on the MunitionDetonation interaction for AAR purposes.

5.2.1.1 Direct Fire

For direct fire engagements, the federate initiating the weapons engagement will populate the TargetObjectIdentifier parameter with the object ID of the intended target. As described above, the target federate will adjudicate damage to the applicable target object based on the DetonationResultCode value and munitions type.

5.2.1.2 Indirect Fire

For indirect fire engagements, the federate initiating the weapons engagement will leave the TargetObjectIdentifier parameter blank. The munitions impact and detonate at the location dictated by the DetonationLocation parameter with effects determined by federate(s) owning objects within the bursting area or radius of effects of the munitions depending on the MunitionType.

5.2.2 Consumption

The level of detail associated with consumption in JLVC is dependant on the presence of JDLM as a federate. In the absence of JDLM, other federates, e.g. JCATS, AWSIM, etc. consume some classes of supplies based on object type and supply class. While the consumption rates may be very accurate for the particular entity object, consumption levels may be under-reported as rates are aggregated for higher level units since all unit equipment may not be represented. For example, fuel consumption for attack and reconnaissance helicopters in a cavalry squadron may be accurately tracked while the High Mobility Multipurpose Wheeled Vehicles (HMMWVs) in the Headquarters troop may not be represented at all. If JDLM is not included in the federation, response cell personnel should therefore increase fuel consumption reports by 10%-15% depending on unit type and representation.

If JDLM is included in the federation, it more accurately represents supply consumption, in both type and quantity, by the complete unit so adding the additional percentage is unnecessary.

5.2.3 Resigning

The federate shall invoke resignFederationExecution to leave the HLA federation execution.

5.2.4 Crash Recovery

Those federates capable of saving their own state, shall do so every 30 minutes. Thus if crash recovery is necessary, a federate need restore from a state no older than 30 minutes. The federate should then either catch up to the current federation time before rejoining or, if confident that its state is unchanged during the crash period, rejoin.

Before rejoining, a federate shall wait at least three times the maximum interval between heartbeats. In JLVC, this typically equates to three minutes, although it may be more for a particular federate.

6.0 JLVC Interoperability Working Group

6.1 Purpose

The JLVC Interoperability Working Group (JIWG) is the body through which USJFCOM will establish and maintain the integrity of the JLVC Federation. The JIWG is chartered to:

- Define technical architecture and standards for the JLVC Federation.
- Improve reliability and ensure more predictable performance of the JLVC Federation.
- Promote rapid joint training event preparation and execution by supporting JLVC Federation composability with joint and Service developed federates.
- Improve verification and validation (V&V) of systems supporting the JLVC Federation.

The JIWG will support USJFCOM's control of the JLVC Federation while providing a venue for Service participation in JLVC Federation definition and evolution. It also provides a forum for discussing changes to the JLVC Federation that result from the identification of new training requirements, the evolution of technology, or the resolution of problems identified during operational tests and joint training events. The JIWG helps ensure that changes to the JLVC Federation are coordinated between all stakeholders, resulting in the best possible solution for the joint training community.

6.2 Scope

The following represent the JLVC Federation components of primary concern to the JIWG:

- The JLVC FOM, the JLVC RID, the JLVC Order of Battle Service (OBS) XML schema, and TENA Logical Range Object Model (LROM).
- The JLVC federates.
- The underlying distributed simulation infrastructure that supports the JLVC Federation, to include the HLA RTI and TENA middleware.

The JIWG process for maintaining the integrity of the JLVC Federation is related to, but independent from that used to manage the evolution of the individual M&S federates or the HLA and TENA middleware used in the joint training environment. Although USJFCOM exercises full control over configuration management (CM) for the JLVC FOM, RID, OBS schema and LROM, the JLVC federates and simulation infrastructure are so tightly coupled that JLVC Federation releases will be referenced to a specific version of each. While JIWG members may also participate in configuration control boards for these other systems and joint requirements may drive changes to these

systems, the individual components of the JLVC Federation will normally be managed by separate CM bodies and processes.

6.3 Organization

The JIWG provides a forum for assessing requirements, defining and implementing change and periodically reviewing the JLVC Federation baseline. The JIWG is made up of USJFCOM JLVC leadership, USJFCOM and Service systems engineering staff and subject matter experts, JLVC federate application developers, JTEN networking engineers and other subject matter experts as designated by the JIWG Chairperson.

The Technical Development and Innovation Branch (TDIB) Development Section Chief, the Deputy Development Section Chief, the JLVC Federation Manager and the JLVC Federation Architect will form the nucleus of the JIWG. As the JLVC Federation lead, the TDIB Development Section Chief will provide overall guidance and policy for JLVC Federation changes and evolution. The TDIB Deputy Development Section Chief, serving as the JIWG Chairperson, will manage the activities of the JIWG in accordance with guidance from the TDIB Development Section Chief and higher authority. The JLVC Federation Manager will be responsible for the day-to-day CM of the JLVC Federation baseline, while the JLVC Federation Architect will be responsible for formal definition of the JLVC technical architecture and standards.

Other designated personnel from TDIB engineering, test and V&V teams, the Joint Exercise Support Branch (JESB), the Services, and the JLVC federate and RTI developer community will compose the membership of the JIWG. The JIWG Chairperson will approve specific membership in the group. Table 11 identifies the JIWG members (both voting and non-voting), their roles and their responsibilities in the CM process.

Table 11. JLVC Integration Working Group Membership

Member	Role	Voting Status	Responsibilities
TDIB Development Section Chief	JLVC Federation Lead	Voting	Provides overarching authority and direction for JLVC Federation evolution and changes; approves changes.
TDIB Deputy Development Section Chief	JIWG Chairperson	Voting	Oversees functions of the JIWG, assigns members to ensure performance of the CM process, and monitors the change process.
JLVC Federation Manager	JLVC Federation CM Manager, JIWG member	Voting	Plans and directs design, implementation, integration and technical tests for changes to the JLVC baseline, provides quality assurance for the baseline.

Table 11. JLVC Integration Working Group Membership

Member	Role	Voting Status	Responsibilities
JLVC Federation Architect	Subject matter expert for JLVC technical architecture, JIWG member	Voting	Controls JLVC Federation baseline configuration to include technical architecture and standards, performs configuration audits, manages JLVC releases.
JLVC Federate Developers	Subject matter experts on federate development, integration and functionality	Voting	Review proposed changes for technical and cost impacts.
JESD Operational Lead	Represent USJFCOM users of JLVC Federation	Voting	Participates in review of proposed changes and provide guidance.
Service Representatives	Represent Service interests, JIWG members	Voting	Participate in review of proposed changes and provide guidance.
JLVC Federation Test Lead	JIWG member	Advisory	
JTEN Network Engineer	JIWG member	Advisory	
JTDS Data Manager	JIWG member	Advisory	

The JIWG Chairperson assigns members of the JIWG to change requests on a case-by-case basis. The members of the JIWG are thereby task-organized by the JIWG Chairperson to address each particular change request. The JLVC Federation Manager, who is typically assigned to each change request, and the other assigned JIWG members analyze the technical validity, merit, technical impact and cost and schedule impact of the requested change, and provide a proposed plan for implementing the change. The TDIB Development Section Chief approves all changes to the JLVC Federation. After approval, the JLVC Federation Manager under the direction of the JIWG Chairperson, works to implement the required changes in accordance with the approved plan.

7.0 Configuration Management Processes

The purpose of this section is to establish uniform CM practices for the JLVC software development, integration, and delivery processes. CM is the process of identifying and managing components in a system, which includes controlling software releases, documentation, and change management.

The goal is to provide an environment that is accountable yet flexible while keeping low overhead. This plan will allow staff the most latitude to develop existing and new software while maintaining a certain level of accountability.

This plan is not intended to replace individual federate developer's CM systems and plans. It also does not establish, nor intend to establish, ownership of software introduced into the Joint Advanced Training Technologies Laboratory (JATTL) environment. The scope of this plan is strictly to identify the software and its basic attributes in the JLVC Federation and JATTL environment.

7.1 CM System

Most SW engineers working either in the JATTL or at their home site currently use a development CM system to manage development efforts. This plan will not replace those individual CM systems but establish an independent external system to track versions and high level description of changes and enhancements.

Currently Microsoft Excel spreadsheets or PowerPoint slides are used to capture versions of releasable software packages. The version tracking spreadsheet will be maintained on the "P" drive at P:\Configuration Management\CM Tracker.

7.2 Federation Changes

Changes to the Federation are submitted via the following:

- Federation Change Request (FCR): any software requirement that applies to, enhances, or impacts the entire federation architecture not just one specific system.

- Multi-Purpose Report (MPR): The MPR replaces the legacy Software Trouble Report (STR) and Enhancement Change Proposal (ECP) records. An MPR will be used to request enhancements to software and to report a software product that does not work as intended.

7.3 Version Control

MPRs will result in a version release. Version releases shall be categorized as *Spiral*, *Maintenance,* or *Patch*. Each of these releases meets specific criteria.

- Spiral release – This is the planned semi-annual or annual release of the JLVC Federation. The first number in the version number indicates the spiral or major version release i.e. the "2" in V **2**.1.
- Maintenance release – This is intended to fix any shortfalls in a Spiral release. The maintenance release will contain only fixes to identified problems in the current spiral. Maintenance versions will be identified by the second number in the version i.e. the "1" in V 2.**1**.
- Patch release – this is an immediate response to fix an urgent or critical problem identified in the federation. Patches will be identified as the third number in the version i.e. the second "1" in V 2.1.**1**.

The timing of a version release depends on the planning and criticality of the issue. Spiral and Interim releases are planned events and follow the formal development and testing processes. Maintenance releases and patches are dependant on the situation and criticality at that time. The following guidelines for identifying priority of problems in the Federation are summarized in **Error! Reference source not found.**.

Table 12. STR Resolution and Response

SEVERITY	STR RATING	DESCRIPTION	RESPONSE
CRITICAL	**1**	Problem - Defect crashes software or makes software unstable. Mission cannot be accomplished with current defect.	Immediate fix - TD bypasses development testing. Patch is issued to operational test environment
CRITICAL	**2**	Problem - Defect is preventing architecture from performing a mission critical function and there is no work-around.	Immediate fix - TD bypasses development testing. Patch is issued to operational test environment
PRIORITY	**3**	Problem is preventing the architecture from performing a mission critical function and there is a valid work-around	Create a fix and test in development environment. Maintenance release created once patch is verified.
ROUTINE	**4**	Problem or fault is preventing a non-mission critical function.	Fixes are incorporated into maintenance releases.
ROUTINE	**5**	Problem with mismatch between capability and documentation - works but not as advertised.	Fixes are incorporated into maintenance releases.

Three versions of the JLVC Federation shall be maintained. Those versions are the *Spiral*, *Maintenance,* and *Development* versions.

- *Spiral Version* – JLVC baseline software is considered to be Released Version after a Spiral development test event such as JLVC V3.0. The software packages for the JLVC Federation will be on the download website after a major JLVC test.

- *Maintenance Version (*and *Patches)* - Subsequent maintenance releases of the JLVC software packages will be available after follow-on test events as required. These will have updated software that addresses identified problems included in this release. Maintenance Version will be released following a JLVC test.
- *Development Version* - A development version will be maintained while in use on the JLVC testbed.

7.4 Software Information

Elements of information to capture, at a minimum, will include:

- Name of software being captured
- Name of developer
- Name of company
- Date checked in
- Version number
- High-level description of changes/updates

Documentation required for software packages:

- Software Version Description Document
- Training Manual/User Manual
- Technical Manuals

7.5 Development and Test Events

During development and test events, information about the software introduced into the JATTL environment shall be captured using the aforementioned CM spreadsheet. Software information shall be captured at the beginning and the end of each event in order to show what the federation (or other environment) started and ended with. This also enables the development branch to publish configuration-managed versions of the software to a downloadable website for use by external customers.

Information shall be captured in accordance with the following procedures.

7.5.1 Development Event Procedure

For a development event, the following steps will be followed to accomplish a minimum level of CM:

1. Developer that will submit federate software for inclusion in the event will fill out part I of an Event Questionnaire.

2. Upon arrival, or before, developer will check software into CM system through the CM manager.

3. During event developer tests code, makes changes to enhance or fix problems as needed.

4. At conclusion of event developer checks software through the CM manager, as it is, into CM system again.

5. Developer fills out part II of the Event Questionnaire.

6. If required, after event conclusion, developer checks “executable” version of software into CM system through the CM manager as the official released version.

7. Developer is responsible to ensure associated documentation is checked into the CM System.

7.5.2 Test Event Procedure

For a test event, the following steps will be followed to accomplish a minimum level of CM:

1. Developers submitting new versions of software will check the software into CM system through the CM manager before start of test event.

2. If changes to software are required during test event, fill out part I of Event Questionnaire. Changes are only to be made if errors are catastrophic to federate and/or federation.

3. Upon completion of test event, confirm version of software in CM system.

If changes to software were made during test event:

1. Check software into CM system through the CM manager.

2. Complete part II of Event Questionnaire.

3. If required, after event conclusion, developer checks “executable” version of software into CM system through the CM manager as the official released version.

4. Developer is responsible to ensure associated documentation is checked into the CM system.

APPENDIX A: Acronyms

Table 13. Acronyms

Term/Acronym	Definition
AADC	Area Air Defense Commander
AAR	After Action Review
AC	Austere Challenge
ACE-IOS	Air and Space Constructive Environment Information Operations Simulation
ADOCS	Air Defense Operations Center System
ADSI	Air Defense Systems Integrator
AFAMS	Air Force Agency for Modeling and Simulation
AFATDS	Advanced Field Artillery Tactical Data System
AFB	Air Force Base
AFD	Assumed Friend
AMDWS	Air and Missile Defense Workstations
AOR	Area of Responsibility
ASAS	All Source Analysis System
AS/NE	Ardent Sentry – Northern Edge
ATO	Air Tasking Order
AWACS	Airborne Warning And Control System
AWSIM	Air Warfare Simulation
BFTT	Battle Force Tactical Trainer
BOM	Base Object Model
C	Constructive
C2	Command and Control
C2PC	Command and Control Personal Computer
C4I	Command, Control, Communications, Computers and Intelligence
CDT	Complex Data Type
CENTCOM	U.S. Central Command
CJTFEX	Commander Joint Task Force Exercise
CM	Configuration Management

Table 13. Acronyms

Term/Acronym	Definition
CMETL	Core Mission Essential Task List
COCOM	Combatant Command
CONUS	Continental United States
COP	Common Operational Picture
CPU	Central Processing Unit
DCEE	Distributed Continuous Experimentation Environment
DDG	Guided Missile Destroyer
DDM	Data Distribution Management
DIF	Data Interchange Format
DIS	Distributed Interactive Simulation
DISA	Defense Information Systems Agency
DM	Declaration Management
DMOC	Distributed Mission Operations Center
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DR	Dead Reckoning
DRM	Dead Reckoning Model
EDCSS	Environmental Data Cube Support System
EP3	Electronic P-3 Orion
ESG	Environmental Scenario Generator
EUCOM	European Command
FED	Federation Execution Data
FEDEP	Federation Development and Execution Process
FedExec	Federation Execution Process
FMS-D	Flight Mission Simulator – Digital
FOM	Federation Object Model
FRD	Blue Friend
GALE Lite	Generic Area Limitation Environment Lite
GCC	Geocentric Coordinate System

Table 13. Acronyms

Term/Acronym	Definition
GCCS	Global Command and Control System
GCCS-J	Global Command and Control System – Joint
GIAC	Graphical Input Aggregate Control
GOTH	Gateway of TENA/HLA
GRIM	Guidance, Rationale and Interoperability Model
GUI	Graphical User Interface
HLA	High Level Architecture
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HUMINT	Human Intelligence
IEEE	Institute of Electrical & Electronics Engineers, Inc.
IMINT	Information Management Intelligence
IOS	Internet Operating System
IP	Internet Protocol
ISR	Intelligence, Surveillance, and Reconnaissance
ITM	IBM Tivoli Monitoring
JATTL	Joint Advanced Training Technologies Laboratory
JCATS	Joint Conflict and Tactical Simulation
JDAARS	Joint Distributed After Action Review System
JDLM	Joint Deployment Logistics Module
JECG	Joint Exercise Control Group
JECS	Joint Exercise Control Station
JESB	Joint Exercise Support Branch
JICO	Joint Interface Control Officer
JIWG	JLVC Interoperability Working Group
JLOD	JCATS Low Overhead Driver
JLVC	Joint Live, Virtual and Constructive
JLVC ID	JLVC Identifier
JMECS	Joint MSEL and Control System
JNTC	Joint National Training Capability

Table 13. Acronyms

Term/Acronym	Definition
JRTC-AWII	Joint Readiness Training Center – Air Warrior II
JSAF	Joint Semi-Automated Forces
JSPA	Joint Simulation Protocol Analyzer
JSWS	Joint Services Work Station
JTASC	Joint Training, Analysis, and Simulations Center
JTDS	Joint Training Data Services
JTEN	Joint Training and Experimentation Network
JTF	Joint Task Force
JTLS	Joint Theater Level Simulation
JUNIT	Joint Unit
JWFC	Joint Warfighting Center
L	Live
LAN	Local Area Network
LISP	LISt Processing (language)
LRC	Local Runtime Infrastructure Component
LROM	Logical Range Object Model
LVC	Live, Virtual and Constructive
M&S	Modeling & Simulation
MC02	Millennium Challenge 2002
METL	Mission Essential Task List
METOC	Meteorological Operations
MILES	Multiple Integrated Laser Engagement System
MIL-STD	Military Standard
MOM	Management Object Model
MSEL	Master Scenario Events List
MTC	Multi-TADIL Capability
MUSE	Multiple Unified Simulation Environment
NIC	Network Interface Card
NORAD	North American Aerospace Defense Command

Table 13. Acronyms

Term/Acronym	Definition
NORTHCOM	U.S. Northern Command
NSC ERF	National Simulation Center Entity Resolution Federation
NTC	National Training Center
NTF	NATO Training Federation
NWARS NG	National Wargaming Simulation Next Generation
NWDC	Navy Warfare Development Command
OBS	Order of Battle Services
OCONUS	Outside Continental United States
OMD	Object Model Data
OMDT	Object Model Development Tool
OMT	Object Model Template
OPTASK	Operational Tasking
OS	Operating System
OS-OTG	Operating System – Over The Horizon Targeting Gold
OVLY2	Overlay 2
OVLY3	Overlay 3
PACOM	Pacific Command
PPLI	Precise Participant Location and Identification
R2	Reporting Responsibility
RID	RTI Initialization Data
RPR	Real-time Platform Reference
RTI	Run-time Infrastructure
RtiExec	RTI Executive Process
SIGINT	Signals Intelligence
SIMPLE	Simulation to C4I Interchange Module for Plans, Logisitics and Exercises
SISO	Simulation Interoperability Standards Organization
SOF	Special Operations Forces
SOS	System of Systems
STRATCOM	U.S. Strategic Command

Table 13. Acronyms

Term/Acronym	Definition
TACSIM	Tactical Simulation
TADIL	Tactical Digital Information Link
TBMCS	Theater Battle Management Core System
TCP/IP	Transmission Control Protocol/Internet Protocol
TDIB	Technical Development and Innovation Branch
TDL	Tactical Data Link
TENA	Test and Training Enabling Architecture
TENCAP	Tactical Exploitation of National Capabilities
TF	Terminal Fury
TGS	Terrain Generation Server
TQ	Track Quality
TRANSCOM	U.S. Transportation Command
TS	Talisman Saber
UE	United Endeavor
UK	United Kingdom
UML	Unified Modeling Language
USJFCOM	United States Joint Forces Command
USMTF	United States Message Text Format
V	Virtual
V&V	Verification and Validation
VLAN	Virtual Local Area Network
VTC	Virtual Technology Corporation
WAN	Wide Area Network
WRC	Western Range Complex
XCTC	Exportable Combat Training Capability
XML	eXtensible Markup Language

APPENDIX B: References

B.1 General

To Be Added

B.2 Directives

1. DODD 5000.59, DoD Modeling and Simulation (M&S) Management
2. DODI 5000.61, DoD Modeling and Simulation (M&S) Verification, Validation and Accreditation (VV&A)
3. CJCSI 8510.01A, Joint Modeling and Simulation Management
4. DoD High Level Architecture (HLA) Memorandum of Agreement (MOA)
5. DoD Modeling and Simulation (M&S) Steering Committee Memo on Strategic Vision for DoD M&S

B.3 Technical

1. High Level Architecture (HLA) - High-Level Architecture Interface Specification, Version 1.3, U.S. Department of Defense, 2 April 1998.
2. Distributed Interactive Simulation (DIS) – IEEE Standard 1278.1a-1998, IEEE Standard for Distributed Interactive Simulation.
3. Test and Training Enabling Architecture (TENA) – TENA Architecture Reference Document, version 2002.
4. Over-the-Horizon Targeting GOLD – Operational Specification for Over-the-Horizon Targeting GOLD (OS-OTG), Baseline 2004
5. Link 16 (Link 16) – MIL STD 6016C, DoD Interface Standard – Tactical Data Link (TDL) 16 Message Standard
6. United States Message Text Format (USMTF) – MIL STD 6040, DoD Interface Standard – U.S. Message Test Formatting Program, Baseline 2008.
7. SISO-REF-010-2006
8. Experimentation with DDM schemes_, D. Coffin, M. Calef, D. Macanucco, and W. Civinskas, Spring 1999 Simulation Interoperability Workshop, 99S-SIW-053
9. Guidance, Rationale, and Interoperability Modalities for the Real-time Platform Reference Federation Object Model (RPR FOM), Version 1.0, Simulation Interoperability Standards Organization, 10 September 1999

APPENDIX C: JLVC Federation Version 3.1 Configuration

The JLVC Federation Version 3.1 Configuration appears in Table 14. The information is current as of December 2009.

Table 14. JLVC Configuration (version 3.1)

Component	Version Number
JCATS	9.0.3
JCATS Bridge	9.0.4 Patch 4
JLOD	1.0.3
JECS	3.1.0
JMECS	3.1.0
JMEM	3.1.0
JRC	3.1.0
JAWS	3.1.0
JSPA	3.1.0
JMECS-NS	3.1.0
JSAF	4.0.3
JBUS	4.0.3
SAR	2.0.6
AWSIM	4.3.0.0
GIAC	2.29 P5
RGI	RGI_229_P5
SGS	5.6.2.1.2
ACE-CSI	4.3.0.0
ACE-IOS	2.8.11.1
SIMPLE	4.0.0j-5
NWARS-NG	1.4
JDLM	JLVC SP3.1 28.01.00
HLA Listener	R01.03.00006
MUSE/AFSERS	8.4.0.3
TIU	IC 2.0 E6

Table 14. JLVC Configuration (version 3.1)

Component	Version Number
HDC	3.1.4
EDCSS	1.2.1
FOM	20 Mar 09
XML	1.4

APPENDIX D: Order of Battle Service Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XMLSpy v2008 sp1 (http://www.altova.com) by Bruce Robbins (Joint
Warfare Center)
-->
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="unqualified" attributeFormDefault="unqualified">
- <xs:element name="UGUdata">
- <xs:annotation>
<xs:documentation>Describes report content.</xs:documentation>
</xs:annotation>
- <xs:complexType>
- <xs:sequence>
<xs:element ref="SideList" />
<xs:element ref="UnitList" minOccurs="0" />
</xs:sequence>
- <xs:attribute name="SchemaVersion" type="xs:string">
- <xs:annotation>
<xs:documentation>The version of the OBS Schema that the instance XML file is
using.</xs:documentation>
</xs:annotation>
</xs:attribute>
- <xs:attribute name="ScenarioVersion" type="xs:string">
- <xs:annotation>
<xs:documentation>The version of the scenario that this document is
describing.</xs:documentation>
</xs:annotation>
</xs:attribute>
- <xs:attribute name="ScenarioCreationDate" type="xs:dateTime">
- <xs:annotation>
<xs:documentation>The date on which this scenario was created.</xs:documentation>
</xs:annotation>
</xs:attribute>
- <xs:attribute name="XMLCreationDate" type="xs:dateTime">
- <xs:annotation>
<xs:documentation>The date on which thie instance XML file was
created.</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
- <xs:element name="SideList">
- <xs:annotation>
<xs:documentation>List of sides</xs:documentation>
</xs:annotation>
```

```
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="Side" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="Side">
- <xs:annotation>
  <xs:documentation>Instance of a side</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="SideName" />
  <xs:element ref="Relationship" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="Color" type="xs:string" use="optional" />
  <xs:attribute name="DISCountryCode" type="xs:int" use="optional" />
  <xs:attribute name="OTHGoldCountryCode" type="xs:string" use="optional" />
    </xs:complexType>
    </xs:element>
- <xs:element name="SideName" nillable="false">
- <xs:annotation>
  <xs:documentation>Unique name of a side or affiliation.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:ID" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="Relationship">
- <xs:annotation>
  <xs:documentation>A relation of one side to another</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="SideReference" />
  <xs:element ref="Relation" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="SideReference" nillable="true">
- <xs:annotation>
  <xs:documentation>Reference to the side that the unit is affiliated
     with.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:IDREF" />
    </xs:simpleType>
```

```
    </xs:element>
- <xs:element name="Relation">
- <xs:annotation>
  <xs:documentation>Type of relation between sides</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="FRIEND" />
  <xs:enumeration value="HOSTILE" />
  <xs:enumeration value="NEUTRAL" />
  <xs:enumeration value="UNKNOWN" />
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
- <xs:element name="UnitList">
- <xs:annotation>
  <xs:documentation>List of units.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="Unit" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="Unit">
- <xs:annotation>
  <xs:documentation>Instance of a unit.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="EntityID" />
- <xs:choice>
  <xs:element ref="SideReference" />
  <xs:element ref="TOESuperior" />
    </xs:choice>
  <xs:element ref="AttachmentSuperior" minOccurs="0" />
  <xs:element ref="Position" minOccurs="0" />
  <xs:element ref="CombatSystemsList" minOccurs="0" />
  <xs:element ref="ELSData" minOccurs="0" />
    </xs:sequence>
  <xs:attribute name="InstanceName" type="xs:string" use="required" />
  <xs:attribute name="OwningFederateName" type="xs:string" use="required" />
- <xs:attribute name="Domain" use="required">
- <xs:annotation>
  <xs:documentation source="SISO-REF-10-2005 25 March 2005">The physical domain
    in which the Unit operates. Domain List is from the DIS Domain
    enumeration</xs:documentation>
```

```
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="Land" />
  <xs:enumeration value="Air" />
  <xs:enumeration value="Surface" />
  <xs:enumeration value="Subsurface" />
  <xs:enumeration value="Space" />
  <xs:enumeration value="Other" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="Echelon" use="required">
- <xs:annotation>
  <xs:documentation>Unit Echelon. Use OTH-GOLD echelon list. Same as
    JCATS.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="Air Army" />
  <xs:enumeration value="Air Command" />
  <xs:enumeration value="Air Control Party" />
  <xs:enumeration value="Air Corps" />
  <xs:enumeration value="Air Detachment" />
  <xs:enumeration value="Air Division" />
  <xs:enumeration value="Air Element" />
  <xs:enumeration value="Air Flight" />
  <xs:enumeration value="Air Group" />
  <xs:enumeration value="Air Regiment" />
  <xs:enumeration value="Air Squadron" />
  <xs:enumeration value="Air Wing" />
  <xs:enumeration value="Army" />
  <xs:enumeration value="Army Group" />
  <xs:enumeration value="Battalion" />
  <xs:enumeration value="Battery" />
  <xs:enumeration value="Border District Headquarters" />
  <xs:enumeration value="Brigade" />
  <xs:enumeration value="Combat Command" />
  <xs:enumeration value="Command" />
  <xs:enumeration value="Company" />
  <xs:enumeration value="Corps" />
  <xs:enumeration value="Detachment" />
  <xs:enumeration value="Division" />
  <xs:enumeration value="Divisional Artillery Group" />
  <xs:enumeration value="Fleet" />
  <xs:enumeration value="Front" />
  <xs:enumeration value="Group" />
  <xs:enumeration value="Group of Forces" />
```

```
<xs:enumeration value="Group of Fronts" />
<xs:enumeration value="Komendatura" />
<xs:enumeration value="Major Fleet" />
<xs:enumeration value="National Defense Headquarters" />
<xs:enumeration value="Naval Detachment" />
<xs:enumeration value="Naval Division" />
<xs:enumeration value="Naval Force" />
<xs:enumeration value="Naval Group" />
<xs:enumeration value="Naval Section" />
<xs:enumeration value="Naval Squadron" />
<xs:enumeration value="Numbered Fleet" />
<xs:enumeration value="Otryad" />
<xs:enumeration value="Patrol" />
<xs:enumeration value="Platoon" />
<xs:enumeration value="Regiment" />
<xs:enumeration value="Regimental Artillery Group" />
<xs:enumeration value="Region" />
<xs:enumeration value="Section" />
<xs:enumeration value="Squad" />
<xs:enumeration value="Squadron" />
<xs:enumeration value="Task Element" />
<xs:enumeration value="Task Element, Abbreviated" />
<xs:enumeration value="Task Force" />
<xs:enumeration value="Task Force, Abbreviated" />
<xs:enumeration value="Task Group" />
<xs:enumeration value="Task Group, Abbreviated" />
<xs:enumeration value="Task Unit" />
<xs:enumeration value="Task Unit, Abbreviated" />
<xs:enumeration value="Team" />
<xs:enumeration value="Theater Army" />
<xs:enumeration value="Troop" />
<xs:enumeration value="Zastrova" />
<xs:enumeration value="Unknown" />
  </xs:restriction>
  </xs:simpleType>
  </xs:attribute>
- <xs:attribute name="UIC" type="xs:string" use="required">
- <xs:annotation>
  <xs:documentation>Unit Identification Code. This is assigned to Battalion Level Units
    and generated for Units below Battaliion Level</xs:documentation>
  </xs:annotation>
  </xs:attribute>
- <xs:attribute name="FBCB2URN" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>Unit Reference Number. This is the communication code assigned
    to the Unit. This is assigned from the FBCB2 database or generated when not
    playing with FBCB2 communication nets</xs:documentation>
  </xs:annotation>
```

```
    </xs:attribute>
  <xs:attribute name="MountedOnUnit" type="xs:IDREF" use="optional" />
  <xs:attribute name="MilSTD2525BSymbolName" type="xs:string" use="optional" />
- <xs:attribute name="UnitAggregateTemplate" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>Assigned by JCATS, it represents a pointer to Unit aggregation
    data. It is NOTa graphic symbol name</xs:documentation>
    </xs:annotation>
    </xs:attribute>
  <xs:attribute name="UnitShortName" type="xs:string" use="optional" />
- <xs:attribute name="UnitCrestFileName" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>The graphic data file name for the Unit Crest displayed on the UGU
    tree.</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="Faction" type="xs:string">
- <xs:annotation>
  <xs:documentation>A subset of Side. A Side can have many Factions. This
    implementation will be replaced by a tree structure wrt Side as a later
    date</xs:documentation>
    </xs:annotation>
    </xs:attribute>
    </xs:complexType>
    </xs:element>
- <xs:element name="TOESuperior" nillable="true">
- <xs:annotation>
  <xs:documentation>Reference to the unit identified to be the superior of this unit or
    combat system.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:IDREF" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="GeoPosition">
- <xs:annotation>
  <xs:documentation>The StartEx position of the Unit. The Lat/Lon as a real number.
    Unit Radians</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="Latitude" />
  <xs:element ref="Longitude" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="ReferencePosition">
- <xs:annotation>
```

```
  <xs:documentation>A position that is the reference point specified by Offset
     position.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="Latitude" />
  <xs:element ref="Longitude" />
- <xs:element name="Orientation">
- <xs:simpleType>
- <xs:restriction base="xs:decimal">
  <xs:minInclusive value="0" />
  <xs:maxInclusive value="6.28318530717958647692528676655 9" />
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="OffsetPosition">
- <xs:annotation>
  <xs:documentation>The X-Y offest from a specified Reference Position in
     meters</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
- <xs:element name="XOffset" default="0">
- <xs:simpleType>
  <xs:restriction base="xs:double" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="YOffset" default="0">
- <xs:simpleType>
  <xs:restriction base="xs:double" />
    </xs:simpleType>
    </xs:element>
  <xs:element ref="EntityReference" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
- <xs:element name="ELSData">
- <xs:annotation>
  <xs:documentation>Additional data to initialize the ELS</xs:documentation>
    </xs:annotation>
- <xs:complexType>
  <xs:attribute name="PrototypeName" type="xs:string" use="required" />
  <xs:attribute name="TargetCategory" type="xs:string" use="optional" />
  <xs:attribute name="TargetSubcategory" type="xs:string" use="optional" />
- <xs:attribute name="UnitType" use="required">
```

```
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="GROUND" />
  <xs:enumeration value="NAVAL" />
  <xs:enumeration value="SQUADRON" />
  <xs:enumeration value="DEPOT" />
  <xs:enumeration value="FARP" />
  <xs:enumeration value="AIRBASE" />
  <xs:enumeration value="HRU" />
  <xs:enumeration value="" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
  <xs:attribute name="CommandableFlag" type="xs:boolean" use="required" />
- <xs:attribute name="Formation" use="required">
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="LINE" />
  <xs:enumeration value="COLUMN" />
  <xs:enumeration value="VEE" />
  <xs:enumeration value="WEDGE" />
  <xs:enumeration value="CIRCLE" />
  <xs:enumeration value="MOB" />
  <xs:enumeration value="" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
    </xs:complexType>
    </xs:element>
- <xs:element name="CombatSystemsList">
- <xs:annotation>
  <xs:documentation>List of combat systems</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="CombatSystem" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="CombatSystem">
- <xs:annotation>
  <xs:documentation>Instance of a combat system</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="EntityID" />
  <xs:element ref="AttachmentSuperior" minOccurs="0" />
```

```
- <xs:choice>
  <xs:element ref="Equipment" />
  <xs:element ref="Personnel" />
    </xs:choice>
  <xs:element ref="DISCode" minOccurs="0" />
  <xs:element ref="NetList" minOccurs="0" />
  <xs:element ref="Position" minOccurs="0" />
    </xs:sequence>
- <xs:attribute name="ClassName" type="xs:string" use="required">
- <xs:annotation>
  <xs:documentation source="JCATS SDB 1.2.5">The JCATS Class
     Name</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="MobilityType" use="required">
- <xs:annotation>
  <xs:documentation source="JCATS SDB 1.2.5">SIMPLE Mobility
     Type</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="DISMOUNT" />
  <xs:enumeration value="WHEELED VEHICLE" />
  <xs:enumeration value="TRACKED VEHICLE" />
  <xs:enumeration value="FIXED WING" />
  <xs:enumeration value="ROTARY WING" />
  <xs:enumeration value="WATERCRAFT" />
  <xs:enumeration value="SUBMARINE" />
  <xs:enumeration value="STATIONARY" />
  <xs:enumeration value="" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="Role" type="xs:string" use="required">
- <xs:annotation>
  <xs:documentation>The rolel this Combat Systems has in scenario. Should be unique
     in Scenario</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="FBCB2URN" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation source="Unit Reference Number assigned by FBCB2 or generated by
     UGU">FBCB2 Unit Reference Number. Assigned by hand or
     generated</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="FBCB2Equipped" type="xs:boolean" use="optional">
- <xs:annotation>
```

```
  <xs:documentation>Is this Combat System capable of FBCB2 data
    comm</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="FBCB2ContinuousFeed" type="xs:boolean" use="optional">
- <xs:annotation>
  <xs:documentation>Is this a continous source of FBCB2 data. ex: UAC. Entered by
    hand</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="BFTEquipped" type="xs:boolean" use="optional">
- <xs:annotation>
  <xs:documentation>Is this Combat System capable of partitipating in Blue Force
    Tracker</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="MTSEquipped" type="xs:boolean" use="optional">
- <xs:annotation>
  <xs:documentation>Is this Combat System capabole of participating in Movement
    Tracking System</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="JTLSCSName" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>The JTLS Combat System name</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="MilStd2525BSymbol" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>Mil Std 2525B Symbol String representing the Combat
    System</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="VisualizationSymbolName" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation>3D symbol name for MUSE/AFSERS and other 3D Visualization
    systems</xs:documentation>
    </xs:annotation>
    </xs:attribute>
- <xs:attribute name="CombatSystemIconName">
- <xs:annotation>
  <xs:documentation>2D symbol name for display in OBS or other tools sharing OBS
    graphics</xs:documentation>
    </xs:annotation>
    </xs:attribute>
    </xs:complexType>
    </xs:element>
- <xs:element name="Equipment">
- <xs:annotation>
```

```
  <xs:documentation>Non-life form combat system.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="CrewList" minOccurs="0" />
  <xs:element ref="PassengerList" minOccurs="0" />
    </xs:sequence>
  <xs:attribute name="NIIN" type="xs:long" use="required" />
  <xs:attribute name="FSN" type="xs:int" use="required" />
  <xs:attribute name="LIN" type="xs:string" use="required" />
  <xs:attribute name="PrimaryWeaponLIN" type="xs:string" use="optional" />
  <xs:attribute name="NeedsCrew" type="xs:boolean" use="optional" />
    </xs:complexType>
    </xs:element>
- <xs:element name="CrewList">
- <xs:annotation>
  <xs:documentation>List of assigned crew.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="CrewID" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="CrewID" nillable="false">
- <xs:annotation>
  <xs:documentation>Reference to the combat system defining the individual crew
     member.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:IDREF" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="PassengerList">
- <xs:annotation>
  <xs:documentation>List of combat systems that are designated as passengers for the
     combat system they are listed under.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="PassengerID" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="PassengerID" nillable="true">
- <xs:annotation>
```

```
  <xs:documentation>Reference to the combat system defining the individual
    passenger.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:IDREF" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="Personnel">
- <xs:annotation>
  <xs:documentation>Life form.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
  <xs:attribute name="Rank" type="xs:string" use="required" />
  <xs:attribute name="MOS" type="xs:string" use="required" />
  <xs:attribute name="Billet" type="xs:string" use="required" />
- <xs:attribute name="PersonnelType" use="required">
- <xs:annotation>
  <xs:documentation>Indicates location status of the personnel. C - Crew A - Assistant P
    - Passenger D - Dismount</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="C" />
  <xs:enumeration value="A" />
  <xs:enumeration value="P" />
  <xs:enumeration value="D" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="Gender" use="optional">
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="M" />
  <xs:enumeration value="F" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="BloodType" use="optional">
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="O POS" />
  <xs:enumeration value="O NEG" />
  <xs:enumeration value="A POS" />
  <xs:enumeration value="A NEG" />
  <xs:enumeration value="B POS" />
  <xs:enumeration value="B NEG" />
  <xs:enumeration value="AB POS" />
  <xs:enumeration value="AB NEG" />
```

```
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="ReligiousAffiliation" use="optional">
- <xs:simpleType>
  <xs:restriction base="xs:string" />
    </xs:simpleType>
    </xs:attribute>
    </xs:complexType>
    </xs:element>
- <xs:element name="PersonnelType">
- <xs:annotation>
  <xs:documentation>Indicates the location status of the personnel.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="C" />
  <xs:enumeration value="A" />
  <xs:enumeration value="P" />
  <xs:enumeration value="D" />
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
- <xs:element name="DISCode">
- <xs:annotation>
  <xs:documentation>The Distributed Interactive Simulation (DIS) enumeration code
    for the Combat System.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
  <xs:attribute name="Kind" type="xs:int" use="required" />
  <xs:attribute name="Domain" type="xs:int" use="required" />
  <xs:attribute name="Country" type="xs:int" use="required" />
  <xs:attribute name="Category" type="xs:int" use="required" />
  <xs:attribute name="Subcategory" type="xs:int" use="required" />
  <xs:attribute name="Specific" type="xs:int" use="required" />
  <xs:attribute name="Extra" type="xs:int" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="NetList">
- <xs:annotation>
  <xs:documentation>The list of Communication Nets in which the Combat System
    Participates</xs:documentation>
    </xs:annotation>
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="Net" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="size" type="xs:int" use="required" />
```

```
    </xs:complexType>
    </xs:element>
- <xs:element name="Net">
- <xs:annotation>
  <xs:documentation>An instance of a communication Net</xs:documentation>
    </xs:annotation>
- <xs:complexType>
  <xs:attribute name="Name" type="xs:string" use="required" />
  <xs:attribute name="Priority" type="xs:int" use="required" />
- <xs:attribute name="Type" use="optional">
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="CMD" />
  <xs:enumeration value="FIRES" />
  <xs:enumeration value="LOG" />
  <xs:enumeration value="INTEL" />
  <xs:enumeration value="FBCB2" />
  <xs:enumeration value="BF" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="CommDevice" type="xs:string" use="optional">
- <xs:annotation>
  <xs:documentation source="Communication Device Name - The equipment designation
    used for this comm link ex: SINGARS" />
    </xs:annotation>
    </xs:attribute>
    </xs:complexType>
    </xs:element>
- <xs:element name="Aggregate">
- <xs:annotation>
  <xs:documentation>Element containing required aggregate
    information.</xs:documentation>
    </xs:annotation>
- <xs:complexType>
  <xs:attribute name="AggregateSymbolName" type="xs:string" use="required" />
  <xs:attribute name="ClassificationSymbolName" type="xs:string" use="required" />
- <xs:attribute name="AggMobilityType" use="required">
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="Ground" />
  <xs:enumeration value="FixedWing" />
  <xs:enumeration value="WaterOnly" />
  <xs:enumeration value="Helicopter" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
- <xs:attribute name="FormationType" use="required">
```

```
- <xs:simpleType>
- <xs:restriction base="xs:string">
  <xs:enumeration value="Assembly" />
  <xs:enumeration value="Column" />
  <xs:enumeration value="Line" />
  <xs:enumeration value="Vee" />
  <xs:enumeration value="Wedge" />
    </xs:restriction>
    </xs:simpleType>
    </xs:attribute>
  <xs:attribute name="FormUpRadius" type="xs:double" use="required" />
  <xs:attribute name="OffSetDistance" type="xs:double" use="required" />
  <xs:attribute name="MountRadius" type="xs:double" use="required" />
  <xs:attribute name="PartialDefiladeFraction" type="xs:double" use="required" />
  <xs:attribute name="FullDefiladeFraction" type="xs:double" use="required" />
    </xs:complexType>
    </xs:element>
- <xs:element name="EntityID" nillable="false">
- <xs:annotation>
  <xs:documentation>Unique identifier of the unit or combat system</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:ID" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="AttachmentSuperior">
- <xs:annotation>
  <xs:documentation>Reference to the unit identified to be the unit the obect is
     assigned or attached to.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:IDREF" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="EntityReference" nillable="false">
- <xs:annotation>
  <xs:documentation>Reference to an existing entity. This could be a unit or combat
     system.</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
  <xs:restriction base="xs:ID" />
    </xs:simpleType>
    </xs:element>
- <xs:element name="Latitude">
- <xs:annotation>
  <xs:documentation>A real number Units: Radians. Range: -Pi to Pi</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
```

```
- <xs:restriction base="xs:decimal">
  <xs:minInclusive value="-3.14159265358979323846264338327 95" />
  <xs:maxInclusive value="3.14159265358979323846264338327 95" />
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
- <xs:element name="Longitude">
- <xs:annotation>
  <xs:documentation>A real number Units: Radians. Range: 0-2pi</xs:documentation>
    </xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:decimal">
  <xs:minInclusive value="0" />
  <xs:maxInclusive value="6.28318530717958647692528676655 9" />
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
- <xs:element name="Position">
- <xs:complexType>
- <xs:choice>
  <xs:element ref="GeoPosition" />
  <xs:element ref="ReferencePosition" />
  <xs:element ref="OffsetPosition" />
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:schema>
```

APPENDIX E: Federation Object Model Publication and Subscription Set – Object Classes

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.AggregateEntity	AggregateMarking							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	AggregateState							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	Dimensions							X	X	X	X										
BaseEntity.AggregateEntity	EntityIdentifier							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	EntityIdentifiers							X	X			X	X	X	X						
BaseEntity.AggregateEntity	EntityType							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	ForceIdentifier							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	Formation							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	IsPartOf							X	X			X	X	X	X						
BaseEntity.AggregateEntity	JLVCID							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	LiveVirtualConstructive							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	NumberOfSilentEntities							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.AggregateEntity	NumberOfVariableDatums							X	X			X	X	X	X						
BaseEntity.AggregateEntity	RelativeSpatial											X	X	X	X						
BaseEntity.AggregateEntity	Spatial							X	X	X	X	X	X	X	X						
BaseEntity.AggregateEntity	SubAggregateIdentifiers							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	AcousticSignatureIndex							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	AlternateEntityType							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	AlternateForceIdentifier							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	CamouflageType					X		X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	DamageState					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	EngineSmokeOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	EntityIdentifier					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	EntityType					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	ExternalLightsOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	FirePowerDisabled					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	FlamesPresent							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.CulturalFeature	ForceIdentifier					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	HasFuelSupplyCap							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	HasRecoveryCap							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	HasRepairCap							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	Immobilized					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	InfraredSignatureIndex							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	InternalHeatSourceOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	InternalLightsOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	IsConcealed					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	IsPartOf							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	JLVCID							X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	LiveEntityMeasuredSpeed											X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	LiveVirtualConstructive					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	Marking					X		X	X	X	X	X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.CulturalFeature	PeriscopeUp							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	PowerPlantOn					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	RelativeSpatial											X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	SmokePlumePresent							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	SnorkelUp							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	Spatial					X		X	X	X	X	X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	TentDeployed							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.CulturalFeature	TrailingEffectsCode					X		X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Lifeform.Human	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	AlternateEntityType							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Lifeform.Human	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Lifeform.Human	CamouflageType					X	X	X	X			X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Lifeform.Human	ComplianceState							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	DamageState					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	EmitterDamage											X	X	X	X						
BaseEntity.PhysicalEntity.Lifeform.Human	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Lifeform.Human	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	EntityIdentifier					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	EntityType					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	FirePowerDisabled					X	X	X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	FlashLightsOn							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	ForceIdentifier					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	HasRepairCap							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Lifeform.Human	Immobilized					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	IsConcealed					X	X	X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	JLVCID							X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	LiveVirtualConstructive					X	X	X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	Marking					X	X	X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	PowerPlantOn					X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	PrimaryWeaponState							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	SecondaryWeaponState							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	SmokePlumePresent							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Lifeform.Human	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	Spatial					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	StanceCode							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Lifeform.Human	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Lifeform.Human	TrailingEffectsCode					X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	AlternateEntityType							X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Munition	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Munition	CamouflageType			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	DamageState			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	EntityIdentifier			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	EntityType			X		X		X	X			X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Munition	FirePowerDisabled			X		X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	FlamesPresent			X				X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	ForceIdentifier			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	HasRepairCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	Immobilized			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	IsConcealed			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	JLVCID			X				X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	LauncherFlashPresent																		X		X
BaseEntity.PhysicalEntity.Munition	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	LiveVirtualConstructive			X		X		X	X			X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Munition	Marking			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	PowerPlantOn			X		X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	SmokePlumePresent			X				X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	Spatial			X		X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Munition	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Munition	TrailingEffectsCode			X		X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	AlternateEntityType							X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	AntiCollisionLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Aircraft	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Platform.Aircraft	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.Aircraft	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.Aircraft	CamouflageType	X		X	X	X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	DamageState	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	EmitterDamage											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.Aircraft	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.Aircraft	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	EntityIdentifier	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	EntityType	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	FirePowerDisabled	X		X	X	X		X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	FlamesPresent			X				X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	ForceIdentifier	X		X	X	X		X	X	X	X	X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Aircraft	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HasFuelSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HasRecoveryCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HasRepairCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	Immobilized	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	InteriorLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	IsConcealed	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	JLVCID	X		X	X			X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	LauncherRaised							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Aircraft	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	LiveVirtualConstructive	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	Marking	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	Mission_Number			X								X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	PowerPlantOn	X		X	X	X		X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	SmokePlumePresent			X				X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	Spatial	X		X	X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Aircraft	SpotLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Aircraft	TailLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Aircraft	TrailingEffectsCode	X		X	X	X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	AlternateEntityType							X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	CBRNEffectLowRes							X	X												X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.GroundVehicle	CamouflageType					X	X	X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	DamageState					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	EntityIdentifier					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	EntityType					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	FirePowerDisabled					X	X	X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	ForceIdentifier					X	X	X	X	X	X	X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HasFuelSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HasRecoveryCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HasRepairCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	Immobilized					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	InteriorLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle	IsConcealed					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	JLVCID							X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	LiveVirtualConstructive					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	Marking					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	PeriscopeUp							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle	PowerPlantOn					X	X	X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	Spatial					X	X	X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	TailLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle	TrailingEffectsCode					X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	AlternateEntityType							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	CBRNEffectLowRes							X	X												X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	CamouflageType				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	DamageState				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	EntityIdentifier				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	EntityType				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	FirePowerDisabled				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	ForceIdentifier				X	X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HasRepairCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	Immobilized				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	InteriorLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	IsConcealed				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	JLVCID				X			X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	LiveVirtualConstructive				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	Marking				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	PeriscopeUp							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	PowerPlantOn				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	Spatial				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	TailLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.AirDefense	TrailingEffectsCode				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	AlternateEntityType							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	CBRNEffectLowRes							X	X												X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	CamouflageType				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	DamageState				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	EmitterDamage											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	EntityIdentifier				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	EntityType				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	FirePowerDisabled				X	X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	ForceIdentifier				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HasRepairCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	Immobilized				X	X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	InteriorLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	IsConcealed				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	JLVCID				X			X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	LiveVirtualConstructive				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	Marking				X	X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	PowerPlantOn				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	Spatial				X	X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	TailLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.GroundVehicle.SPArtillery	TrailingEffectsCode				X	X	X	X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	AlternateEntityType							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Platform.Spacecraft	CBRNProtection							X	X												X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Spacecraft	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.Spacecraft	CamouflageType					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	DamageState					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	EntityIdentifier					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	EntityType					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	FirePowerDisabled					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	ForceIdentifier					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	HasRepairCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	HatchState							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Spacecraft	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	Immobilized					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	InteriorLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	IsConcealed					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	JLVCID							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	LiveVirtualConstructive					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	Marking					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	PowerPlantOn					X		X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Platform.Spacecraft	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	Spatial					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	TailLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Spacecraft	TrailingEffectsCode					X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Submersib	AlternateEntityType							X	X	X	X	X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
leVessel																					
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	CamouflageType				X	X		X	X			X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	DamageState				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.Submersib	EmitterDamage											X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
leVessel																					
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	EntityIdentifier				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	EntityType				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	FirePowerDisabled				X	X		X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	ForceIdentifier				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	FormationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Submersib	HasFuelSupplyCap							X	X	X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
leVessel																					
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	HasRecoveryCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	HasRepairCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	Immobilized				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	InteriorLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	IsConcealed				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Submersib	JLVCID				X			X	X	X	X	X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
leVessel																					
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	LiveVirtualConstructive				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	Marking				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	PowerPlantOn				X	X		X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.Submersib	RampDeployed							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
leVessel																					
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	Spatial				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	TailLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SubmersibleVessel	TrailingEffectsCode				X	X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVe	AcousticSignatureIndex							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	AfterburnerOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	AlternateEntityType							X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	AntiCollisionLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	BlackOutBrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	BrakeLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	CallSign											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.SurfaceVe	CamouflageType				X	X		X	X			X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	DamageState				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	EmitterDamage											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	EntityIdentifier				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	EntityType				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	FirePowerDisabled				X	X		X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	ForceIdentifier				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVe	FormationLightsOn							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HasFuelSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HasRecoveryCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HasRepairCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HatchState							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	HeadLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	Immobilized				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	InteriorLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVe	IsConcealed				X	X		X	X	X	X	X	X	X	X				X		X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	JLVCID				X			X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	LandingLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	LauncherRaised							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	LiveVirtualConstructive				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	Marking				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	NavigationLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVe	PowerPlantOn				X	X		X	X	X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	RampDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	RelativeSpatial											X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	RunningLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	Spatial				X	X		X	X	X	X	X	X	X	X				X		X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	SpotLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	TailLightsOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Platform.SurfaceVe	TentDeployed							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
ssel																					
BaseEntity.PhysicalEntity.Platform.SurfaceVessel	TrailingEffectsCode				X	X		X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	AlternateEntityType							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	CBRNEffectLowRes							X	X												X
BaseEntity.PhysicalEntity.Radio	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Radio	CamouflageType							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	DamageState							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	EmitterDamage											X	X	X	X						
BaseEntity.PhysicalEntity.Radio	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Radio	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	EntityIdentifier							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	EntityType							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	FirePowerDisabled							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Radio	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	ForceIdentifier							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	HasAmmunitionSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	HasFuelSupplyCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	HasRecoveryCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	HasRepairCap							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	Immobilized							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	IsConcealed							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	IsPartOf							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	JLVCID							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	LiveVirtualConstructive							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	Marking							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	PeriscopeUp							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Radio	PowerPlantOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	Spatial							X	X			X	X	X	X		X				X
BaseEntity.PhysicalEntity.Radio	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Radio	TrailingEffectsCode							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	AcousticSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	AlternateEntityType							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	AlternateForceIdentifier							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	CBRNProtection							X	X												X
BaseEntity.PhysicalEntity.Sensor	CamouflageType				X			X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	DamageState				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	EmitterDamage											X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Sensor	EmitterStatus											X	X	X	X						
BaseEntity.PhysicalEntity.Sensor	EngineSmokeOn							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	EntityIdentifier				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	EntityType				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	FirePowerDisabled				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	FlamesPresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	ForceIdentifier				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	HasAmmunitionSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	HasFuelSupplyCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	HasRecoveryCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	HasRepairCap							X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	Immobilized				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	InfraredSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	IsConcealed				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	IsPartOf							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Sensor	JLVCID				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	LiveVirtualConstructive				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	Marking				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	MissionKill																				X
BaseEntity.PhysicalEntity.Sensor	PeriscopeUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	PowerPlantOn				X			X	X	X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	SmokePlumePresent							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	SnorkelUp							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	Spatial				X			X	X	X	X	X	X	X	X		X				X
BaseEntity.PhysicalEntity.Sensor	TentDeployed							X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Sensor	TrailingEffectsCode				X			X	X			X	X	X	X						X
BaseEntity.PhysicalEntity.Supplies	AcousticSignatureIndex							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Supplies	AlternateEntityType							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	AlternateForceIdentifier							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	CamouflageType							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	DamageState							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	EngineSmokeOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	EntityIdentifier							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	EntityType							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	FirePowerDisabled							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	FlamesPresent							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	ForceIdentifier							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	HasAmmunitionSupplyCap							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	HasFuelSupplyCap							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	HasRecoveryCap							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	HasRepairCap							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	Immobilized							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.Supplies	InfraredSignatureIndex							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	IsConcealed							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	IsPartOf							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	JLVCID							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	LiveEntityMeasuredSpeed											X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	LiveVirtualConstructive							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	Marking							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	PeriscopeUp							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	PowerPlantOn							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	RadarCrossSectionSignatureIndex							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	SmokePlumePresent							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	SnorkelUp							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	Spatial							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	TentDeployed							X	X			X	X	X	X						
BaseEntity.PhysicalEntity.Supplies	TrailingEffectsCode							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.SurfaceInstallation	AcousticSignatureIndex											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	AlternateEntityType									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	AlternateForceIdentifier											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	CamouflageType											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	DamageState									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	EngineSmokeOn											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	EntityIdentifier			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	EntityType			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	FirePowerDisabled									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	FlamesPresent											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	ForceIdentifier			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	HasAmmunitionSupplyCap									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	HasFuelSupplyCap									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	HasRecoveryCap									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	HasRepairCap									X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.SurfaceInstallation	Immobilized									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	InfraredSignatureIndex											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	IsConcealed									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	IsPartOf											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	JLVCID			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	LiveEntityMeasuredSpeed											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	LiveVirtualConstructive			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	Marking			X						X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	PeriscopeUp											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	PowerPlantOn									X	X	X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	RadarCrossSectionSignatureIndex											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	RelativeSpatial											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	SmokePlumePresent											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	SnorkelUp											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	Spatial			X						X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BaseEntity.PhysicalEntity.SurfaceInstallation	TentDeployed											X	X	X	X						X
BaseEntity.PhysicalEntity.SurfaceInstallation	TrailingEffectsCode											X	X	X	X						X
BasicEncyclopedia.BE_Facility	BE_Number											X	X	X	X		X				
BasicEncyclopedia.BE_Facility	FUN											X	X	X	X		X				
BasicEncyclopedia.BE_Facility	O_Suffix											X	X	X	X		X				
BasicEncyclopedia.BE_Facility	categoryCode											X	X	X	X		X				
BasicEncyclopedia.BE_Facility	centerLocation											X	X	X	X		X				
BasicEncyclopedia.BE_Facility	structuralStatus											X	X	X	X		X				
BasicEncyclopedia.DMPI	BE_Facility											X	X	X	X		X				
BasicEncyclopedia.DMPI	DMPI_ID											X	X	X	X		X				
BasicEncyclopedia.DMPI	DMPI_Status											X	X	X	X		X				
BasicEncyclopedia.DMPI	FUN											X	X	X	X		X				
BasicEncyclopedia.DMPI	centerLocation											X	X	X	X		X				
EmbeddedSystem.EmitterSystem	EmitterFunctionCode				X			X	X			X	X	X	X						X
EmbeddedSystem.EmitterSystem	EmitterIndex				X			X	X			X	X	X	X		X				X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.EmitterSystem	EmitterType				X			X	X			X	X	X	X		X				X
EmbeddedSystem.EmitterSystem	EntityIdentifier				X			X	X			X	X	X	X		X				X
EmbeddedSystem.EmitterSystem	EventIdentifier							X	X			X	X	X	X						X
EmbeddedSystem.EmitterSystem	HostObjectIdentifier				X			X	X			X	X	X	X		X				X
EmbeddedSystem.EmitterSystem	LiveVirtualConstructive							X	X			X	X	X	X						X
EmbeddedSystem.EmitterSystem	RelativePosition				X			X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	AlternateMode4			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	BeamAzimuthCenter							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	BeamAzimuthSweep							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	BeamElevationCenter							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	BeamElevationSweep							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	BeamSweepSync							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	EmergencyOn			X				X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	EntityIdentifier			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	EventIdentifier			X				X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	HostObjectIdentifier			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	IdentSquawkFlashOn			X				X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Layer2DataAvailable							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	LiveVirtualConstructive					X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode1Code			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode1Enabled			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode1IsDamaged			X		X	X	X	X	X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode1IsMalfunctioning			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode1IsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode2Code			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode2Enabled			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode2IsDamaged			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode2IsMalfunctioning			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode2IsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode3ACode			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode3AEnabled			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode3AIsDamaged			X		X	X	X	X	X	X	X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode3AIsMalfunctioning			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode3AIsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4Enabled			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4IsDamaged			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4IsMalfunctioning			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4IsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4PseudoCrypto			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode4PseudoCryptoAvailable			X				X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CAltitude			X		X	X	X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CAltitudeAvailable			X				X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CEnabled			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CIsDamaged			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CIsMalfunctioning			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	Mode5CIsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	ModeSEnabled							X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	ModeSIsDamaged							X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	ModeSIsMalfunctioning							X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	ModeSIsOn							X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	ModeSIsTcasI							X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	RelativePosition			X		X	X	X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SecondaryOperationalDataParameter1							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SecondaryOperationalDataParameter2							X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	StiOn					X	X	X	X			X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SystemIsOn			X	X	X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SystemIsOperational			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SystemMode			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SystemName			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder	SystemType			X		X	X	X	X	X	X	X	X	X	X						X
EmbeddedSystem.RadioTransmitter	AntennaPatternType							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	CryptoSystem							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	CryptographicMode							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.RadioTransmitter	EncryptionKeyIdentifier							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	EntityIdentifier							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	Frequency							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	FrequencyBandwidth							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	HighOrderFreq							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	HostObjectIdentifier							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	LiveVirtualConstructive							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	RFModulationSystemType							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	RadioIndex							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	RadioInputSource							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	RadioSystemType							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	RelativePosition							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	StreamTag							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	TimeHopInUse							X	X			X	X	X	X						
EmbeddedSystem.RadioTransmitter	TransmittedPower							X	X			X	X	X	X		X				

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmbeddedSystem.RadioTransmitter	TransmitterOperationalStatus							X	X			X	X	X	X		X				
EmbeddedSystem.RadioTransmitter	WorldLocation							X	X			X	X	X	X						
EmitterBeam.JammerBeam	BeamAzimuthCenter							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamAzimuthSweep							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamElevationCenter							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamElevationSweep							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamFunctionCode							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamIdentifier							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	BeamParameterIndex							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	EffectiveRadiatedPower							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	EmissionFrequency							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	EmitterSystemIdentifier							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	EventIdentifier							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	FrequencyRange							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	HighDensityJam							X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmitterBeam.JammerBeam	JammedObjectIdentifiers							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	JammerTechnique							X	X												X
EmitterBeam.JammerBeam	JammingModeSequence							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	PulseRepetitionFrequency							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	PulseWidth							X	X			X	X	X	X						X
EmitterBeam.JammerBeam	SweepSynch							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamAzimuthCenter							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamAzimuthSweep							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamElevationCenter							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamElevationSweep							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamFunctionCode			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamIdentifier			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	BeamParameterIndex			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	EffectiveRadiatedPower			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	EmissionFrequency			X	X			X	X			X	X	X	X						X

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
EmitterBeam.RadarBeam	EmitterSystemIdentifier			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	EventIdentifier			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	FrequencyRange			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	HighDensityTrack							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	PulseRepetitionFrequency			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	PulseWidth			X	X			X	X			X	X	X	X						X
EmitterBeam.RadarBeam	SweepSynch							X	X			X	X	X	X						X
EmitterBeam.RadarBeam	TrackObjectIdentifiers			X	X			X	X			X	X	X	X						X
Engineering.Crater	disGuise							X	X			X	X	X	X						
Engineering.Crater	force							X	X			X	X	X	X						
Engineering.Crater	geometry							X	X			X	X	X	X						
Engineering.Crater	guise							X	X			X	X	X	X						
Engineering.Crater	marking							X	X			X	X	X	X						
Engineering.Ditch	disGuise							X	X			X	X	X	X						
Engineering.Ditch	force							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
Engineering.Ditch	geometry							X	X			X	X	X	X						
Engineering.Ditch	guise							X	X			X	X	X	X						
Engineering.Ditch	marking							X	X			X	X	X	X						
Engineering.FloatingBarrier	disGuise							X	X			X	X	X	X						
Engineering.FloatingBarrier	force							X	X			X	X	X	X						
Engineering.FloatingBarrier	geometry							X	X			X	X	X	X						
Engineering.FloatingBarrier	guise							X	X			X	X	X	X						
Engineering.FloatingBarrier	marking							X	X			X	X	X	X						
Engineering.FoxHole	disGuise							X	X			X	X	X	X						
Engineering.FoxHole	force							X	X			X	X	X	X						
Engineering.FoxHole	geometry							X	X			X	X	X	X						
Engineering.FoxHole	guise							X	X			X	X	X	X						
Engineering.FoxHole	marking							X	X			X	X	X	X						
Engineering.Minefields	ActiveStatus							X	X			X	X	X	X						
Engineering.Minefields	Lane							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
Engineering.Minefields	MinefieldAppearanceType							X	X			X	X	X	X						
Engineering.Minefields	disGuise							X	X			X	X	X	X						
Engineering.Minefields	force							X	X			X	X	X	X						
Engineering.Minefields	geometry							X	X			X	X	X	X						
Engineering.Minefields	guise							X	X			X	X	X	X						
Engineering.Minefields	marking							X	X			X	X	X	X						
Engineering.Rubble	disGuise							X	X			X	X	X	X						
Engineering.Rubble	force							X	X			X	X	X	X						
Engineering.Rubble	geometry							X	X			X	X	X	X						
Engineering.Rubble	guise							X	X			X	X	X	X						
Engineering.Rubble	marking							X	X			X	X	X	X						
Engineering.SandBag	disGuise							X	X			X	X	X	X						
Engineering.SandBag	force							X	X			X	X	X	X						
Engineering.SandBag	geometry							X	X			X	X	X	X						
Engineering.SandBag	guise							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
Engineering.SandBag	marking							X	X			X	X	X	X						
Engineering.SubmergedBarrier	disGuise							X	X			X	X	X	X						
Engineering.SubmergedBarrier	force							X	X			X	X	X	X						
Engineering.SubmergedBarrier	geometry							X	X			X	X	X	X						
Engineering.SubmergedBarrier	guise							X	X			X	X	X	X						
Engineering.SubmergedBarrier	marking							X	X			X	X	X	X						
Engineering.VehicleFortification	disGuise							X	X			X	X	X	X						
Engineering.VehicleFortification	force							X	X			X	X	X	X						
Engineering.VehicleFortification	geometry							X	X			X	X	X	X						
Engineering.VehicleFortification	guise							X	X			X	X	X	X						
Engineering.VehicleFortification	marking							X	X			X	X	X	X						
Engineering.VehicleHole	disGuise							X	X			X	X	X	X						
Engineering.VehicleHole	force							X	X			X	X	X	X						
Engineering.VehicleHole	geometry							X	X			X	X	X	X						
Engineering.VehicleHole	guise							X	X			X	X	X	X						

		ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD_Server		NWARS		RGI		SIMPLE	
CLASSNAME	ATTRIBUTE	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
Engineering.VehicleHole	marking							X	X			X	X	X	X						
Engineering.Wire	disGuise							X	X		X	X	X	X	X						
Engineering.Wire	force							X	X		X	X	X	X	X						
Engineering.Wire	geometry							X	X		X	X	X	X	X						
Engineering.Wire	guise							X	X		X	X	X	X	X						
Engineering.Wire	marking							X	X		X	X	X	X	X						

APPENDIX F: Federation Object Model Publication and Subscription Set – Interactions

	ABS		AWSIM		GOTH		JCATS		JDLM		JLOD_Client		JLOD-Server		NWARS		RGI		Simple	
	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
BattleDamage	X						X	X	X	X	X	X	X	X				X		
BattleDamageRepair	X	X					X	X	X	X								X		
DamageAssessment					X		X	X												
EntityControl							X	X	X	X	X	X	X	X						
LogAssessmentRequest							X	X	X	X	X	X	X	X						
LogEntityCreation							X	X	X	X										
LogHealthUpdate							X	X	X	X	X	X	X	X						
LogisticsAssessment			X				X	X	X	X	X	X	X	X						
LogisticsResupply				X			X	X	X	X	X	X	X	X						
LogisticsResupplyControl									X	X										
LogisticsResupplyRequest									X	X										
MovementOrders											X	X	X	X						
MunitionDetonation			X	X	X		X	X	X	X	X	X	X	X				X		
OrgStateInteraction.OrgStateV3							X	X	X	X										

OrgUpdateInteraction							X	X		X										
RepairComplete							X	X	X	X	X	X	X	X						
UnitInsertion							X	X	X	X										
WeaponFire			X		X		X	X	X	X	X	X	X	X				X		

APPENDIX G: Federation Object Model

Published Separately. Available by request from Brian Gregg brian.gregg@att.jfcom.mil

APPENDIX H: Software - Event Questionnaire

Part I - To be filled in before Event Start

1. Date:_____

2. Simulation/Federate name:_____ Version:_____

3. Developer:_____ Documentation:_____

4. Goal during this event:

_____

_____

_____

_____

_____

_____

5. Description of Federate status, problems or enhancements prior to event:

_____

_____

_____

_____

_____

_____

Part II - To be filled in at Event End

1. Date:_____ New Version:_____

2. Did you accomplish your goal? If no, state shortfalls/problems remaining:

_____

_____

_____

_____

_____

_____

3. High level description of changes/fixes made:

_____

_____

_____

_____

_____

4. Other:

_____

_____

_____

_____

_____

_____